

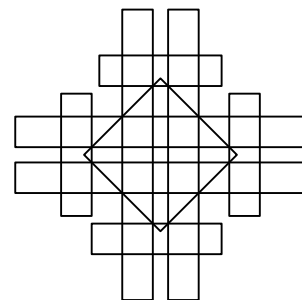
38th International Symposium on Computational Geometry

SoCG 2022, June 7–10, 2022, Berlin, Germany

Edited by

Xavier Goaoc

Michael Kerber



Editors

Xavier Goaoc

LORIA, Université de Lorraine, France
xavier.goaoc@loria.fr

Michael Kerber 

Graz University of Technology, Austria
kerber@tugraz.at

ACM Classification 2012

Theory of computation → Computational geometry; Theory of computation → Design and analysis of algorithms; Mathematics of computing → Combinatorics; Mathematics of computing → Graph algorithms

ISBN 978-3-95977-227-3

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-227-3>.

Publication date

June, 2022

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

License

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0): <https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.SoCG.2022.0

ISBN 978-3-95977-227-3

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

LIPICs – Leibniz International Proceedings in Informatics

LIPICs is a series of high-quality conference proceedings across all fields in informatics. LIPICs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Luca Aceto (*Chair*, Reykjavik University, IS and Gran Sasso Science Institute, IT)
- Christel Baier (TU Dresden, DE)
- Mikolaj Bojanczyk (University of Warsaw, PL)
- Roberto Di Cosmo (Inria and Université de Paris, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Daniel Král' (Masaryk University - Brno, CZ)
- Meena Mahajan (Institute of Mathematical Sciences, Chennai, IN)
- Anca Muscholl (University of Bordeaux, FR)
- Chih-Hao Luke Ong (University of Oxford, GB)
- Phillip Rogaway (University of California, Davis, US)
- Eva Rotenberg (Technical University of Denmark, Lyngby, DK)
- Raimund Seidel (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

■ Contents

Preface	
<i>Xavier Goaoc, Michael Kerber, Aaron T. Becker, Sándor P. Fekete, and Stefan Schirra</i>	0:xi
Conference Organization	
.....	0:xiii
Additional Reviewers	
.....	0:xvii

Regular Papers

Tiling with Squares and Packing Dominos in Polynomial Time	
<i>Anders Aamand, Mikkel Abrahamsen, Thomas Ahle, and Peter M. R. Rasmussen</i>	1:1–1:17
On Cyclic Solutions to the Min-Max Latency Multi-Robot Patrolling Problem	
<i>Peyman Afshani, Mark de Berg, Kevin Buchin, Jie Gao, Maarten Löffler, Amir Nayyeri, Benjamin Raichel, Rik Sarkar, Haotian Wang, and Hao-Tsung Yang</i>	2:1–2:14
On Semialgebraic Range Reporting	
<i>Peyman Afshani and Pingan Cheng</i>	3:1–3:14
Intersection Queries for Flat Semi-Algebraic Objects in Three Dimensions and Related Problems	
<i>Pankaj K. Agarwal, Boris Aronov, Esther Ezra, Matthew J. Katz, and Micha Sharir</i>	4:1–4:14
Twisted Ways to Find Plane Structures in Simple Drawings of Complete Graphs	
<i>Oswin Aichholzer, Alfredo García, Javier Tejel, Birgit Vogtenhuber, and Alexandra Weinberger</i>	5:1–5:18
Edge Partitions of Complete Geometric Graphs	
<i>Oswin Aichholzer, Johannes Obenaus, Joachim Orthaber, Rosna Paul, Patrick Schnider, Raphael Steiner, Tim Taubner, and Birgit Vogtenhuber</i>	6:1–6:16
Minimum-Error Triangulations for Sea Surface Reconstruction	
<i>Anna Arutyunova, Anne Driemel, Jan-Henrik Haunert, Herman Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel, and Heiko Röglin</i>	7:1–7:18
Delaunay-Like Triangulation of Smooth Orientable Submanifolds by ℓ_1 -Norm Minimization	
<i>Dominique Attali and André Lieutier</i>	8:1–8:16
Tighter Bounds for Reconstruction from ϵ -Samples	
<i>Håvard Bakke Bjerkevik</i>	9:1–9:17
Erdős-Szekeres-Type Problems in the Real Projective Plane	
<i>Martin Balko, Manfred Scheucher, and Pavel Valtr</i>	10:1–10:15



True Contraction Decomposition and Almost ETH-Tight Bipartization for Unit-Disk Graphs <i>Sayan Bandyopadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue</i>	11:1–11:16
Unlabeled Multi-Robot Motion Planning with Tighter Separation Bounds <i>Bahareh Banyassady, Mark de Berg, Karl Bringmann, Kevin Buchin, Henning Fernau, Dan Halperin, Irina Kostitsyna, Yoshio Okamoto, and Stijn Slot</i>	12:1–12:16
Optimality of the Johnson-Lindenstrauss Dimensionality Reduction for Practical Measures <i>Yair Bartal, Ora Nova Fandina, and Kasper Green Larsen</i>	13:1–13:16
Quasi-Universality of Reeb Graph Distances <i>Ulrich Bauer, Håvard Bakke Bjerkevik, and Benedikt Fluhr</i>	14:1–14:18
Gromov Hyperbolicity, Geodesic Defect, and Apparent Pairs in Vietoris–Rips Filtrations <i>Ulrich Bauer and Fabian Roll</i>	15:1–15:15
Acute Tours in the Plane <i>Ahmad Biniaz</i>	16:1–16:8
ETH-Tight Algorithms for Finding Surfaces in Simplicial Complexes of Bounded Treewidth <i>Mitchell Black, Nello Blaser, Amir Nayyeri, and Erlend Raa Vågset</i>	17:1–17:16
Asymptotic Bounds on the Combinatorial Diameter of Random Polytopes <i>Gilles Bonnet, Daniel Dadush, Uri Grupel, Sophie Huiberts, and Galyna Livshyts</i>	18:1–18:15
Signed Barcodes for Multi-Parameter Persistence via Rank Decompositions <i>Magnus Bakke Botnan, Steffen Oppermann, and Steve Oudot</i>	19:1–19:18
Dynamic Time Warping Under Translation: Approximation Guided by Space-Filling Curves <i>Karl Bringmann, Sándor Kisfaludi-Bak, Marvin Künnemann, Dániel Marx, and André Nusser</i>	20:1–20:17
Towards Sub-Quadratic Diameter Computation in Geometric Intersection Graphs <i>Karl Bringmann, Sándor Kisfaludi-Bak, Marvin Künnemann, André Nusser, and Zahra Parsaeian</i>	21:1–21:16
Computing Continuous Dynamic Time Warping of Time Series in Polynomial Time <i>Kevin Buchin, André Nusser, and Sampson Wong</i>	22:1–22:16
Long Plane Trees <i>Sergio Cabello, Michael Hoffmann, Katharina Klost, Wolfgang Mulzer, and Josef Tkadlec</i>	23:1–23:17
The Universal ℓ^p -Metric on Merge Trees <i>Robert Cardona, Justin Curry, Tung Lam, and Michael Lesnick</i>	24:1–24:20

On Complexity of Computing Bottleneck and Lexicographic Optimal Cycles in a Homology Class
Erin Wolf Chambers, Salman Parsa, and Hannah Schreiber 25:1–25:15

Parameterized Algorithms for Upward Planarity
Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Robert Ganian, Chrysanthi N. Raftopoulou, and Kirill Simonov 26:1–26:16

Finding Weakly Simple Closed Quasigeodesics on Polyhedral Spheres
Jean Chartier and Arnaud de Mesmay 27:1–27:16

Tight Lower Bounds for Approximate & Exact k -Center in \mathbb{R}^d
Rajesh Chitnis and Nitin Saurabh 28:1–28:15

Flat Folding an Unassigned Single-Vertex Complex (Combinatorially Embedded Planar Graph with Specified Edge Lengths) Without Flat Angles
Lily Chung, Erik D. Demaine, Dylan Hendrickson, and Victor Luo 29:1–29:17

Hop-Spanners for Geometric Intersection Graphs
Jonathan B. Conroy and Csaba D. Tóth 30:1–30:17

Persistent Cup-Length
Marco Contessoto, Facundo Mémoli, Anastasios Stefanou, and Ling Zhou 31:1–31:17

Three-Chromatic Geometric Hypergraphs
Gábor Damásdi and Dömötör Pálvölgyi 32:1–32:13

A Solution to Ringel’s Circle Problem
James Davies, Chaya Keller, Linda Kleist, Shakhar Smorodinsky, and Bartosz Walczak 33:1–33:14

Computing Generalized Rank Invariant for 2-Parameter Persistence Modules via Zigzag Persistence and Its Applications
Tamal K. Dey, Woojin Kim, and Facundo Mémoli 34:1–34:17

Tracking Dynamical Features via Continuation and Persistence
Tamal K. Dey, Michał Lipiński, Marian Mrozek, and Ryan Stechta 35:1–35:17

On the Discrete Fréchet Distance in a Graph
Anne Driemel, Ivor van der Hoog, and Eva Rotenberg 36:1–36:18

Computing a Link Diagram from Its Exterior
Nathan M. Dunfield, Malik Obeidin, and Cameron Gates Rudd 37:1–37:24

On Comparable Box Dimension
Zdeněk Dvořák, Daniel Gonçalves, Abhiruk Lahiri, Jane Tan, and Torsten Ueckerdt 38:1–38:14

Weak Coloring Numbers of Intersection Graphs
Zdeněk Dvořák, Jakub Pekárek, Torsten Ueckerdt, and Yelena Yuditsky 39:1–39:15

ε -Isometric Dimension Reduction for Incompressible Subsets of ℓ_p
Alexandros Eskenazis 40:1–40:14

Short Topological Decompositions of Non-Orientable Surfaces
Niloufar Fuladi, Alfredo Hubard, and Arnaud de Mesmay 41:1–41:16

Robust Radical Sylvester-Gallai Theorem for Quadratics <i>Abhibhav Garg, Rafael Oliveira, and Akash Kumar Sengupta</i>	42:1–42:13
Robust Sylvester-Gallai Type Theorem for Quadratic Polynomials <i>Shir Peleg and Amir Shpilka</i>	43:1–43:15
Swap, Shift and Trim to Edge Collapse a Filtration <i>Marc Glisse and Siddharth Pritam</i>	44:1–44:15
Hardness and Approximation of Minimum Convex Partition <i>Nicolas Grelier</i>	45:1–45:15
Parameterised Partially-Predrawn Crossing Number <i>Thekla Hamm and Petr Hliněný</i>	46:1–46:15
Approximation Algorithms for Maximum Matchings in Geometric Intersection Graphs <i>Sariel Har-Peled and Everett Yang</i>	47:1–47:13
The Complexity of the Hausdorff Distance <i>Paul Jungeblut, Linda Kleist, and Tillmann Miltzow</i>	48:1–48:17
Dynamic Connectivity in Disk Graphs <i>Haim Kaplan, Alexander Kauer, Katharina Klost, Kristin Knorr, Wolfgang Mulzer, Liam Roditty, and Paul Seifert</i>	49:1–49:17
An $(\aleph_0, k + 2)$ -Theorem for k -Transversals <i>Chaya Keller and Micha A. Perles</i>	50:1–50:14
Farthest-Point Voronoi Diagrams in the Presence of Rectangular Obstacles <i>Mincheol Kim, Chanyang Seo, Taehoon Ahn, and Hee-Kap Ahn</i>	51:1–51:15
Point Separation and Obstacle Removal by Finding and Hitting Odd Cycles <i>Neeraj Kumar, Daniel Lokshtanov, Saket Saurabh, Subhash Suri, and Jie Xue</i>	52:1–52:14
A Universal Triangulation for Flat Tori <i>Francis Lazarus and Florent Tallier</i>	53:1–53:18
Sparse Euclidean Spanners with Tiny Diameter: A Tight Lower Bound <i>Hung Le, Lazar Milenković, and Shay Solomon</i>	54:1–54:15
Minimum Height Drawings of Ordered Trees in Polynomial Time: Homotopy Height of Tree Duals <i>Tim Ophelders and Salman Parsa</i>	55:1–55:16
Disjointness Graphs of Short Polygonal Chains <i>János Pach, Gábor Tardos, and Géza Tóth</i>	56:1–56:12
Covering Points by Hyperplanes and Related Problems <i>Zuzana Patáková and Micha Sharir</i>	57:1–57:7
The Degree-Rips Complexes of an Annulus with Outliers <i>Alexander Rolle</i>	58:1–58:14
Chains, Koch Chains, and Point Sets with Many Triangulations <i>Daniel Rutschmann and Manuel Wettstein</i>	59:1–59:18

Nearly-Doubling Spaces of Persistence Diagrams
Donald R. Sheehy and Siddharth S. Sheth 60:1–60:15

From Geometry to Topology: Inverse Theorems for Distributed Persistence
Elchanan Solomon, Alexander Wagner, and Paul Bendich 61:1–61:16

A Positive Fraction Erdős-Szekeres Theorem and Its Applications
Andrew Suk and Ji Zeng 62:1–62:15

Optimal Coreset for Gaussian Kernel Density Estimation
Wai Ming Tai 63:1–63:15

GPU Computation of the Euler Characteristic Curve for Imaging Data
Fan Wang, Hubert Wagner, and Chao Chen 64:1–64:16

Media Exposition

Space Ants: Episode II – Coordinating Connected Catoms
*Julien Bourgeois, Sándor P. Fekete, Ramin Kosfeld, Peter Kramer,
 Benoît Piranda, Christian Rieck, and Christian Scheffer* 65:1–65:6

A Cautionary Tale: Burning the Medial Axis Is Unstable
*Erin Chambers, Christopher Fillmore, Elizabeth Stephenson, and
 Mathijs Wintraecken* 66:1–66:9

Visualizing and Unfolding Nets of 4-Polytopes
Satyan L. Devadoss, Matthew S. Harvey, and Sam Zhang 67:1–67:4

Visualizing WSPDs and Their Applications
Anirban Ghosh, FNU Shariful, and David Wisnosky 68:1–68:4

Subdivision Methods for Sum-Of-Distances Problems: Fermat-Weber Point,
 n-Ellipses and the Min-Sum Cluster Voronoi Diagram
Ioannis Mantas, Evanthia Papadopoulou, Martin Suderland, and Chee Yap 69:1–69:6

An Interactive Framework for Reconfiguration in the Sliding Square Model
Willem Sonke and Jules Wolms 70:1–70:4

CG Challenge

Shadoks Approach to Minimum Partition into Plane Subgraphs
*Loïc Crombez, Guilherme D. da Fonseca, Yan Gerard, and
 Aldo Gonzalez-Lorenzo* 71:1–71:8

Conflict-Based Local Search for Minimum Partition into Plane Subgraphs
Jack Spalding-Jamieson, Brandon Zhang, and Da Wei Zheng 72:1–72:6

Local Search with Weighting Schemes for the CG:SHOP 2022 Competition
Florian Fontan, Pascal Lafourcade, Luc Libralesso, and Benjamin Momège 73:1–73:6

SAT-Based Local Search for Plane Subgraph Partitions
André Schidler 74:1–74:8

■ Preface

The 38th International Symposium on Computational Geometry (SoCG 2022) was held in Berlin, June 7–10, 2022, as part of the Computational Geometry Week (CG Week 2022).

Altogether, 174 papers were submitted to SoCG 2022. After a thorough review process, in which each paper was evaluated by three or more independent reviewers, the program committee accepted 64 papers for presentation at SoCG 2022. These proceedings contain extended abstracts of the accepted papers, limited to 500 lines (excluding references). If any supporting material does not fit in the line limit, the full paper is available at a public repository and referenced in the corresponding extended abstract.

The **Best Paper Award** of SoCG 2022 goes to the paper “Chains, Koch Chains, and Point Sets with many Triangulations” by Daniel Rutschmann and Manuel Wettstein; this paper has been invited to submit an extended version to the Journal of the ACM. The Best Student Presentation Award was determined and announced at the symposium, based on ballots cast by the attendees. A selection of papers were invited to submit an extended version to forthcoming special issues of Discrete & Computational Geometry and the Journal of Computational Geometry dedicated to the symposium.

Two papers, “Robust Sylvester-Gallai type theorem for quadratic polynomials” and “Robust Radical Sylvester-Gallai Theorem for Quadratics”, independently prove the same main result with similar, but not identical techniques. The committee decided to include them both in the proceedings, and have them presented jointly in a single talk at the conference. The final version of each paper provides a comparison with the other paper. The final decision for all other papers was unaffected by the decision to include both papers in the proceedings.

The **SoCG Test of Time Award** of this year goes to the papers “Measuring the Resemblance of Polygonal Curves”, by Helmut Alt and Michael Godau, presented at SoCG 1992, and “Efficient Partition Trees”, by Jirí Matousek, presented at SoCG 1991.

The scientific program of CG Week 2022 was enriched by two distinguished **invited speakers**. An invited talk, entitled “Efficient Querying of Large-Scale Geodata”, was delivered by Hannah Bast from University of Freiburg. A second invited talk, entitled “Computational geometry and topology for spatial structures arising in biology”, was delivered by Heather Harrington from University of Oxford. We thank these plenary speakers for kindly accepting our invitation.

In addition to the technical papers, there were nine submissions to the **multimedia exposition**. The submissions were reviewed, and six of them were accepted for presentation. The extended abstracts that describe these submissions are included in this proceedings volume. The multimedia content can be found at <https://www.computational-geometry.org>.

A continuing feature in this year’s proceedings is the **CG Challenge**, now in its third year being included in the proceedings. The challenge problem this year was to partition a geometric graph in the plane into a small number of planar subgraphs. This year there were 32 teams submitting verified solutions, and these proceedings contain contributions by the four top-placed teams describing their winning approaches.

We thank the authors of all submitted works. We are most grateful to the members of the SoCG Program Committee, the Media Exposition Committee and the CG Challenge Committee for their dedication, expertise, and hard work that ensured the high quality of the works in these proceedings. We are grateful for the assistance provided by the hundreds



of reviewers; without their help it would have been nearly impossible to run the selection process. Finally, we thank Irina Kostitsyna, who kindly accepted to be the Proceedings Chair and did meticulous work.

Many other people contributed to the success of SoCG 2022 and the entire CG Week. We are very grateful to the local organization committee for their work in organizing the event, and to facilitate remote participation. Finally, we thank all the members of the Test of Time Award, Workshop, and Young Researchers Forum Committees, the CG Challenge Advisory Board, and the Computational Geometry Steering Committee.

Xavier Goaoc
SoCG program committee co-chair
Université de Lorraine

Michael Kerber
SoCG program committee co-chair
TU Graz

Aaron T. Becker
Media exposition chair
University of Houston

Sándor P. Fekete
CG challenge co-chair
TU Braunschweig

Stefan Schirra
CG challenge co-chair
Universität Magdeburg

■ Conference Organization

SoCG Program Committee

- Henry Adams, Colorado State University, USA
- SangWon Bae, Kyonggi University, South Korea
- Édouard Bonnet, ENS Lyon, France
- Timothy Chan, University of Illinois at Urbana-Champaign, USA
- Hsien-Chih Chang, Dartmouth College, USA
- Hu Ding, University of Science and Technology, China
- Vida Dujmović, University of Ottawa, Canada
- Herbert Edelsbrunner, IST, Austria
- Radoslav Fulek, UC San Diego, USA
- Xavier Goaoc (co-chair), Université de Lorraine, France
- Stefan Huber, Salzburg University of Applied Sciences, Austria
- Michael Kerber (co-chair), TU Graz, Austria
- Marc van Kreveld, Utrecht University, The Netherlands
- Claudia Landi, Università di Modena, Italy
- Sepideh Mahabadi, Toyota Technological Institute at Chicago, USA
- Yakov Nekrich, Michigan Tech, USA
- Aleksandar Nikolov, University of Toronto, Canada
- Gabriel Nivasch, Ariel University, Israel
- Natan Rubin, Ben-Gurion University, Israel
- Christiane Schmidt, Linköping University, Sweden
- Haitao Wang, Utah State University, USA
- Emo Welzl, ETH Zürich, Switzerland
- Carola Wenk, Tulane University, USA
- Andreas Wiese, Universidad de Chile
- Chee Yap, New York University, USA

SoCG Proceedings Chair

- Irina Kostitsyna, TU Eindhoven, Netherlands

Media Exposition Committee

- Sarel Har-Peled, University of Illinois at Urbana-Champaign, USA
- Sándor Fekete, TU Braunschweig, Germany
- Maarten Löffler, Utrecht University, The Netherlands
- Jason O’Kane, University of South Carolina, USA
- Irene Parada, Technical University of Denmark, Denmark
- Eli Packer, Intel Corporation, Israel
- Brittany Fasy, Montana State University, USA
- Aaron T. Becker (chair), University of Houston, USA

38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber



Leibniz International Proceedings in Informatics

LIPICIS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



CG Challenge Committee

- Sándor Fekete (co-chair), TU Braunschweig, Germany
- Phillip Keldenich, TU Braunschweig, Germany
- Dominik Krupke, TU Braunschweig, Germany
- Stefan Schirra (co-chair), Universität Magdeburg, Germany

CG Challenge Advisory Board

- Bill Cook, University of Waterloo, Canada
- Andreas Fabri, GeometryFactory, France
- Michael Kerber, TU Graz, Austria
- Philipp Kindermann, Universität Würzburg, Germany
- Joe Mitchell, SUNY Stony Brook, USA
- Kevin Verbeek, TU Eindhoven, The Netherlands

SoCG Test of Time Award Committee

- Pankaj K. Agarwal, Duke University, USA
- Dan Halperin, Tel Aviv University, Israel
- Raimund Seidel, Saarland University, Germany

Workshop Committee

- Ulrich Bauer, Technical University of Munich, Germany
- Jie Gao, Rutgers University, USA
- Wouter Meulemans, TU Eindhoven, Netherlands
- David Mount (chair), University of Maryland, USA
- Bei Wang, University of Utah, USA

Young Researchers Forum Program Committee

- Peyman Afshani, Aarhus University, Denmark
- Sergio Cabello, University of Ljubljana, Slovenia
- Hsien-Chih Chang, Dartmouth College, USA
- Anne Driemel (chair), Universität Bonn, Germany
- André Nusser, University of Copenhagen, Denmark
- Zuzana Patáková, Charles University, Czech Republic
- Matias Korman, Mentor Graphics, USA
- Monique Teillaud, INRIA, France

Local Organizing Committee

- Benjamin Aram Berendsohn, Freie Universität Berlin, Germany
- Helena Bergold, Freie Universität Berlin, Germany
- Kenny Chiu, Freie Universität Berlin, Germany
- Aruni Choudhary, Freie Universität Berlin, Germany
- Heike Eckart, Freie Universität Berlin, Germany
- Alexander Kauer, Freie Universität Berlin, Germany

- Katharina Klost, Freie Universität Berlin, Germany
- Kristin Knorr, Freie Universität Berlin, Germany
- Wolfgang Mulzer (chair), Freie Universität Berlin, Germany
- Johannes Obenaus, Freie Universität Berlin, Germany
- Günter Rote, Freie Universität Berlin, Germany
- Max Willert, Freie Universität Berlin, Germany

Steering Committee (2020-2022)

- Mark de Berg (secretary), TU Eindhoven, Netherlands
- Sándor Fekete, TU Braunschweig, Germany
- Michael Hoffmann (chair), ETH Zurich, Switzerland
- Matya Katz, Ben-Gurion University of the Negev, Israel
- Bettina Speckmann, TU Eindhoven, Netherlands
- Yusu Wang (liaison with the Society for Computational Geometry),
University of California San Diego, USA

■ Additional Reviewers

Ahmed Abdelkader	Siu-Wing Cheng	Lee-Ad Gottlieb
Peyman Afshani	Otfried Cheong	Bogdan Grechuk
Hee-Kap Ahn	Samir Chowdhury	Nicolas Grelier
Oswin Aichholzer	David Cohen-Steiner	Joachim Gudmundsson
Elad Aigner-Horev	Éric Colin de Verdière	Andrea Guidolin
Hugo Akitaya	Sebastiano Cultrera	Waldo Gálvez
Ángel Javier Alonso	Justin Curry	Bernd Gärtner
Helmut Alt	Guilherme D. Da Fonseca	Mustafa Hajij
Enrique Alvarado	Gábor Damásdi	Yassine Hamoudi
Sunil Arya	Gautam K Das	Sariel Har-Peled
Ahmad Bilal Asghar	Sandip Das	Herman Haverkort
Stav Ashur	James Davies	Qizheng He
Marco Attene	Arnaud De Mesmay	Teresa Heiss
Sergey Avvakumov	Olivier Devillers	John Hershberger
Martin Balko	Tamal Dey	Robert Hickingbotham
Sayan Bandyapadhyay	Michael Gene Dobbins	Hung Hoang
Yair Bartal	Ondrej Draganov	Michael Hoffmann
Saugata Basu	Anne Driemel	Andreas Holmsen
Ulrich Bauer	Loïc Dubois	Tao Hou
Nicolas Berkouk	Kunal Dutta	Lingxiao Huang
Daniel Bertschinger	Eduard Eiben	Ziyun Huang
Silvia Biasotti	Friedrich Eisenbrand	Thomas Hull
Ahmad Biniaz	Nicolas El Maalouly	Kristof Huszar
Håvard Bakke Bjerkevik	Alex Elchesen	Saeed Ilchi Ghazaan
Prosenjit Bose	David Eppstein	Tanmay Inamdar
Magnus Bakke Botnan	Esther Ezra	R Inkulu
Nicolas Bousquet	Chenglin Fan	Iordan Iordanov
Karl Bringmann	Andreas Emil Feldmann	Lars Jaffke
Robyn Brooks	Stefan Felsner	Shaofeng H.-C. Jiang
Adam Brown	Hendrik Fichtenberger	Alvin Jin
Mickaël Buchet	Christopher Fillmore	Michael Joswig
Kevin Buchin	Arnold Filtser	Dominik Kaaser
Maike Buchin	Omrit Filtser	Farid Karimipour
Boris Bukh	Benedikt Fluhr	Karthik C. S.
Michael Burr	Kyle Fox	Lars Kastner
Johnathan Bush	Stefan Funke	Matthew Katz
Sergio Cabello	Pawel Gawrychowski	Maximilian Katzmann
Wojciech Chachólski	Ishika Ghosh	Phillip Keldenich
Dibyayan Chakraborty	Panos Giannopoulos	Chaya Keller
Parinya Chalermsook	Barbara Giunti	Balázs Keszegh
Erin Chambers	Marc Glisse	Arindam Khan
T-H. Hubert Chan	Rocio Gonzalez-Diaz	Mincheol Kim
Amit Chattophadyay	Jonathan Goodman	Minki Kim
Renjie Chen	Mayank Goswami	Woojin Kim



Sándor Kisfaludi-Bak	Stefan Neumann	Allan Sapucaia
Felix Klesen	Bengt J. Nilsson	Marcus Schaefer
Fabian Klute	Navid Nouri	Christian Scheffer
Dušan Knop	André Nusser	Anna Schenfisch
Benedikt Kolbe	Martin Nöllenburg	Manfred Scheucher
Matias Korman	Joseph O'Rourke	Arne Schmidt
Irina Kostitsyna	Eunjin Oh	Patrick Schnider
Grigorios Koumoutsos	Yoshio Okamoto	Jordan Schupbach
Hana Kourimska	Osman Okutan	Chris Schwiegelshohn
Myroslav Kryven	Tim Ophelders	Martina Scolamiero
Nirman Kumar	Steve Oudot	Mordechai Shalom
Marvin Künnemann	Rasmus Pagh	Vikram Sharma
Elmar Langetepe	Peter Palfrader	Nicholas Sharp
Sylvain Lazard	Fahad Panolan	Chan-Su Shin
Francis Lazarus	Evanthia Papadopoulou	Devansh Shringi
Hung Le	Irene Parada	Anastasios Sidiropoulos
Vadim Lebovici	Salman Parsa	Francesco Silvestri
Dongryeol Lee	Amit Patel	Berit Singer
Michael Lesnick	Pavel Paták	Isabelle Sivignon
Jian Li	Francois Petit	Primoz Skraba
Tongyang Li	Seth Pettie	Michiel Smid
Jyh-Ming Lien	Jeff Phillips	Pablo Soberón
Sunhyuk Lim	Madhusudhan Reddy Pittu	József Solymosi
Jiashuai Lu	Valentin Polishchuk	Bettina Speckmann
Anna Lubiw	Marc Pouget	Jonathan Spreer
Benjamin Lund	Siddharth Pritam	Frank Staals
Hengrui Luo	Ioannis Psarros	Raphael Steiner
Maarten Löffler	Dömötör Pálvölgyi	Elizabeth Stephenson
Sushovan Majhi	Sharath Raghvendra	Miloš Stojaković
Willi Mann	Saladi Rahul	Martin Suderland
Mathieu Mari	Benjamin Raichel	Andrew Suk
Killian Meehan	Rajiv Raman	Yihan Sun
Facundo Memoli	Jan Rataj	Konrad Swanepoel
David L. Millman	Abhishek Rathod	Shuhao Tan
Till Miltzow	Meghana M Reddy	Martin Tancer
Majid Mirzanezhad	Vanessa Robins	Ewin Tang
Guillaume Moroz	Liam Roditty	Erin Taylor
Dmitriy Morozov	Dennis Rohde	Monique Teillaud
David Mount	Alexander Rolle	Francesca Tombari
Michael Moy	Pepijn Edwin Robert	Csaba Tóth
Wolfgang Mulzer	Roos Hoefgeest	Geza Tóth
Elizabeth Munch	Günter Rote	Manuel Trigueros
Tobias Mömke	Cameron Rudd	Konstantinos Tsakalidis
Torsten Mütze	Florian Russold	Takashi Tsuboi
Chie Nara	Daniel Rutschmann	Jan Vahrenhold
Abhinandan Nath	Leonie Ryvkin	Mikael Vejdemo-Johansson
Ofer Neiman	Morteza Saghafian	Kevin Verbeek
Eike Neumann	Michael Sagraloff	Antoine Vigneron

Ziga Virk
Hubert Wagner
Bartosz Walczak
Zhengchao Wan
Bei Wang
Qingsong Wang
Yanhao Wang
Simon Weber
Manuel Wettstein
Max Willert
Mathijs Wintraecken

Sampson Wong
David R. Wood
Matthew Wright
Jie Xue
Sang Duk Yoon
Jingjin Yu
Joshua Zahl
Nicolò Zava
Ji Zeng
Sebastian Zeng
Shira Zerbib

Da Wei Zheng
Ling Zhou
Samson Zhou
Mark de Berg
Sarita de Berg
Stefan de Lorenzo
Vin de Silva
André van Renssen
Tom van der Zanden
Péter Ágoston
Onur Çağırıcı

Tiling with Squares and Packing Dominos in Polynomial Time

Anders Aamand  

MIT, Cambridge, MA, US

Mikkel Abrahamsen  

BARC, University of Copenhagen, Denmark

Thomas Ahle  

BARC, University of Copenhagen, Denmark

Peter M. R. Rasmussen  

BARC, University of Copenhagen, Denmark

Abstract

A polyomino is a polygonal region with axis-parallel edges and corners of integral coordinates, which may have holes. In this paper, we consider planar tiling and packing problems with polyomino pieces and a polyomino container P . We give polynomial-time algorithms for deciding if P can be tiled with $k \times k$ squares for any fixed k which can be part of the input (that is, deciding if P is the union of a set of non-overlapping $k \times k$ squares) and for packing P with a maximum number of non-overlapping and axis-parallel 2×1 dominos, allowing rotations by 90° . As packing is more general than tiling, the latter algorithm can also be used to decide if P can be tiled by 2×1 dominos.

These are classical problems with important applications in VLSI design, and the related problem of finding a maximum packing of 2×2 squares is known to be NP-hard [J. Algorithms 1990]. For our three problems there are known pseudo-polynomial-time algorithms, that is, algorithms with running times polynomial in the *area* or *perimeter* of P . However, the standard, compact way to represent a polygon is by listing the coordinates of the corners in binary. We use this representation, and thus present the first polynomial-time algorithms for the problems. Concretely, we give a simple $O(n \log n)$ -time algorithm for tiling with squares, where n is the number of corners of P . We then give a more involved algorithm that reduces the problems of packing and tiling with dominos to finding a maximum and perfect matching in a graph with $O(n^3)$ vertices. This leads to algorithms with running times $O(n^3 \frac{\log^3 n}{\log^2 \log n})$ and $O(n^3 \frac{\log^2 n}{\log \log n})$, respectively.

2012 ACM Subject Classification Theory of computation \rightarrow Design and analysis of algorithms

Keywords and phrases packing, tiling, polyominoes

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.1

Related Version *Full Version*: <https://arxiv.org/abs/2011.10983>

Funding *Anders Aamand*: Supported by a DFF-International Postdoc Grant from the Independent Research Fund Denmark.

Mikkel Abrahamsen: Supported by Starting Grant 1054-00032B from the Independent Research Fund Denmark under the Sapere Aude research career programme. BARC is supported by the VILLUM Foundation grant 16582.

Thomas Ahle: BARC is supported by the VILLUM Foundation grant 16582.

Peter M. R. Rasmussen: BARC is supported by the VILLUM Foundation grant 16582.

1 Introduction

A chessboard has been mutilated by removing two diagonally opposite corners, leaving 62 squares. Philosopher Max Black asked in 1946 whether one can place 31 dominos of size 1×2 so as to cover all of the remaining squares? Tiling problems of this sort are popular in



© Anders Aamand, Mikkel Abrahamsen, Thomas Ahle, and Peter M. R. Rasmussen; licensed under Creative Commons License CC-BY 4.0

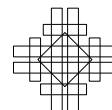
38th International Symposium on Computational Geometry (SoCG 2022).

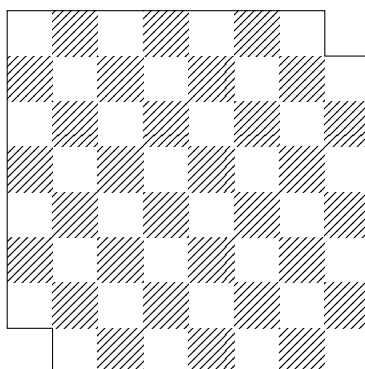
Editors: Xavier Goaoc and Michael Kerber; Article No. 1; pp. 1:1–1:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** The chessboard polyomino envisioned by Max Black.

recreational mathematics, such as the mathematical olympiads¹ and have been discussed by Golomb [14] and Gamow and Stern [12]. The mutilated chessboard and the dominos are examples of the type of polygon called a *polyomino*, which is a polygonal region of the plane with axis-parallel edges and corners of integral coordinates. We allow polyominos to have holes.

From an algorithmic point of view, it is natural to ask whether a given (large) polyomino P can be *tilled* by copies of another fixed (small) polyomino Q , which means that P is the union of non-overlapping copies of Q that may or may not be rotated by 90° and 180° . As the answer is often a boring *no*, one can ask more generally for the largest number of copies of Q that can be *packed* into the given container P without overlapping. Algorithms answering this question (for various Q) turn out to have important applications in very large scale integration (VLSI) circuit technology. As a concrete example, Hochbaum and Maass [15] gave the following motivation for their development of a polynomial-time approximation scheme for packing 2×2 squares into a given polyomino P (using the area representation of P , to be defined later).

“For example, 64K RAM chips, some of which may be defective, are available on a rectilinear grid placed on a silicon wafer. 2×2 arrays of such nondefective chips could be wired together to produce 256K RAM chips. In order to maximize yield, we want to pack a maximal number of such 2×2 arrays into the array of working chips on a wafer.”

Although the mentioned amounts of memory are small compared to those of present day technology, the basic principles behind the production of computer memory are largely unchanged, and methods for circumventing defective cells of wafers (the cells are also known as *dies* in this context) is still an active area of research in semiconductor manufacturing [7, 9, 17, 19].

The most important result in tiling is perhaps the combinatorial group theory approach by Conway and Lagarias [8]. Their algorithmic technique is used to decide whether a given finite region consisting of cells in a regular lattice (triangular, square, or hexagonal) can be tiled by pieces drawn from a finite set of tile shapes. Thurston [25] gives a nice introduction to the technique and shows how it can be used to decide if a polyomino without holes can

¹ See e.g. the “hook problem” of the International Mathematical Olympiad 2004.

be tiled by dominos. The running time is $O(a \log a)$, where a is the *area* of P . Pak, Sheffer, and Tassy [22] described an algorithm with running time $O(p \log p)$, where p is the *perimeter* of P .

The problem of *packing* a maximum number of dominos into a given polyomino P was apparently first analyzed by Berman, Leighton, and Snyder [5] who observed that this problem can be reduced to finding a maximum matching of the incidence graph $G(P)$ of the cells in P : There is a vertex for each 1×1 cell in P , and two vertices are connected by an edge if the two cells share a geometrical edge. The graph $G(P)$ is bipartite, so the problem can be solved in $O(a^{3/2})$ time using the Hopcroft–Karp algorithm, where a is the number of cells (i.e., the area of P).

On the flip-side, a number of hardness results have been obtained for simple tiling and packing problems: Beauquier, Nivat, Remila, and Robson [2] showed that if P can have holes, the problem of deciding if P can be tiled by translates of two rectangles $1 \times m$ and $k \times 1$ is NP-complete as soon as $\max\{m, k\} \geq 3$ and $\min\{m, k\} \geq 2$. Pak and Yang [23] showed that there exists a set of at most 10^6 rectangles such that deciding whether a given *hole-free* polyomino can be tiled with translates from the set is NP-complete. Other generalizations have even turned out to be undecidable: Berger [3] proved in 1966 that deciding whether pieces from a given finite set of polyominoes can tile the plane is Turing-complete (interestingly, Wijshoff and van Leeuwen [26] and Beauquier and Nivat [1] gave algorithms for deciding whether a single polyomino tiles the plane). For packing, Fowler, Paterson, and Tanimoto [11] showed already in the early 80s that deciding whether a given number of 3×3 squares can be packed into a polyomino (with holes) is NP-complete, and the result was strengthened to 2×2 squares by Berman, Johnson, Leighton, Shor, and Snyder [4].

As it turns out, for all of the above results, it is assumed that the container P is represented either as a list of the individual cells forming the interior of P or as a list of the boundary cells. We shall call these representations the *area representation* and *perimeter representation*, respectively. The area and perimeter representations correspond to a unary rather than binary representation of integers and the running times of the existing algorithms are thus only pseudo-polynomial. It is much more efficient and compact to represent P by the coordinates of the corners, where the coordinates are represented as binary numbers. This is the way one would usually represent polygons (with holes) in computational geometry: The corners are given in cyclic order as they appear on the boundary of P , one cycle for the outer boundary and one for each of the holes of P . We shall call such a representation a *corner representation*. With a corner representation, the area and perimeter can be exponential in the input size, so the known algorithms which rely on an area or perimeter representation to be polynomial, are in fact exponential when using this more efficient encoding of the input. Problems that are NP-complete in the area or perimeter representation are also NP-hard in the corner representation, but NP-membership does not necessarily follow. In our practical example of semiconductor manufacturing, the corner representation also seems to be the natural setting for the problem: Hopefully, there are only few defective cells to be avoided when grouping the chips, so the total number of corners of the usable region is much smaller than its area.

El-Khechen, Dulieu, Iacono, and Van Omme [10] showed that even using a corner representation for a polyomino P , the problem of deciding if m squares of size 2×2 can be packed into P is in NP. That was not clear before since the naive certificate specifies the placement of each of the m squares, and so, would have exponential length. Beyond this, we know of no other work using the corner representation for polyomino tiling or packing problems.



Our contribution

While the complexity of the problem of packing 2×2 squares into a polyomino P has thus been settled as NP-complete, the complexity of the tiling problem was left unsettled. Tiling and packing are closely connected in this area of geometry, but their complexities can be drastically different. Indeed, we show in Section 3 that it can be decided in $O(n \log n)$ time by a surprisingly simple algorithm whether P can be tiled by $k \times k$ squares for any fixed $k \in \mathbb{N}$ which can even be part of the input. Here, n is the number of corners of P .² With the area and perimeter representations, it is trivial to decide if P can be tiled in polynomial time (see Section 3), but as noted above, using the corner representation, it is not even immediately obvious that the problem is in NP.

In Section 4, we provide and analyze a simple algorithm, which we denote **simple-packer**, that can decide if m dominos (i.e., rectangles of size 1×2 that can be rotated 90°) can be packed in a given polyomino P . The algorithm works by truncating long edges of P , so that the resulting polyomino P'' has area $O(n^4)$. The graph $G(P'')$ induced by the unit square cells constituting P'' can likewise be constructed in time $O(n^4)$. We then use a multiple-sink multiple-source maximum flow algorithm as a black box [6, 13] to find a maximum matching in $G(P'')$, which results in a running time of $O(n^4 \frac{\log^3 n}{\log^2 \log n})$. In order to decide if P can be tiled with dominos, we can instead use a single-source shortest path algorithm [21], with which one can find perfect matchings in bipartite planar graphs [20], and we obtain a slightly better running time of $O(n^4 \frac{\log^2 n}{\log \log n})$. Although the truncation process of reducing the size to $O(n^4)$ is simple, the proof of correctness is nontrivial and requires some structural lemmas on domino packings.

In the full version of this paper, we manage to reduce the domino packing and tiling problems to finding a maximum and perfect matching in a bipartite planar graph G^* with $O(n^3)$ vertices, instead of $O(n^4)$ as for **simple-packer**. We denote this algorithm **fast-packer**. The actual graph G^* can also be constructed in time $O(n^3)$. This reduction relies on the same structural results as are needed for **simple-packer**, but it is however quite a bit more complicated, and many techniques and technical lemmas are required to prove correctness and bound the size of G^* . We obtain running times of $O(n^3 \frac{\log^3 n}{\log^2 \log n})$ and $O(n^3 \frac{\log^2 n}{\log \log n})$ for packing and tiling, respectively. Table 1 summarises the known and new results.

■ **Table 1** Complexities of the four fundamental tiling and packing problems. Here, n is the number of corners of the container P . The algorithm for tiling with squares works for any size $k \times k$.

Shapes	Tiling	Packing
	$O(n^3 \frac{\log^2 n}{\log \log n})$ [This paper]	$O(n^3 \frac{\log^3 n}{\log^2 \log n})$ [This paper]
	$O(n \log n)$ [This paper]	NP-complete [4, 10]

² We assume throughout the paper that we can make basic operations (additions, subtractions, comparisons) on the coordinates in $O(1)$ time. Otherwise, the time complexity of our square tiling algorithm will be $O(nt \log n)$ and the domino packing algorithm will have complexity $O(n^3 t + n^3 \frac{\log^3 n}{\log^2 \log n})$, where t is the time it takes to make one such operation.

Open problems

Many interesting questions remain for slightly more complex shapes than studied in this paper. For instance, polynomial-time algorithms are known for tiling polyominoes with larger rectangles in the area representation [18, 24]. Are there also algorithms in the corner representation? For the problems that are NP-complete in the area representation [2, 16, 23], which are also contained in NP in the corner representation?

Another interesting problem is to design domino tiling and packing algorithms with better running times than our $\tilde{O}(n^3)$ in the corner representation, e.g., near-linear time algorithms. It seems conceivable that the techniques of [22] can lead to such improvements for tiling *simply connected* polyominoes with dominos. Specifically, it is shown that to decide tilability of a simply connected polyomino P , it suffices to check a certain Lipschitz condition on a *height function* defined on the boundary ∂P of P , and that (essentially) this check can be carried out by considering only $O(p)$ pairs of boundary points, where p is the perimeter of P . It is plausible that one could obtain a similar bound on the number of pairs to be checked in terms of n , which would lead to a faster domino tiling algorithm for hole-free polyominoes.

1.1 Our techniques

Tiling with $k \times k$ squares

We sort the corners of the given polyomino P by the x -coordinates and use a vertical sweep-line ℓ that sweeps over P from left to right. The intuition is that the algorithm keeps track of how the tiling looks in the region of P to the left of ℓ if a tiling exists. As ℓ sweeps over P , we keep track of how the tiling pattern changes under ℓ . Each vertical edge of P that ℓ sweeps over causes changes to the tiling, and we must update our data structure accordingly.

Packing dominos

The basic approach of both the **simple-packer** and the **fast-packer** algorithm is to reduce the packing problem in the polyomino P (with n corners) to a maximum matching problem in a graph G^* with only polynomially many vertices and edges. We prove that a maximum matching in G^* corresponds to a maximum packing of dominos in P . The construction of G^* relies on some non-trivial structural results on domino packings.

The algorithm **simple-packer** first sorts the corners by x -coordinates and considers the corners in this order c_1, \dots, c_n . When $x(c_{i+1}) - x(c_i) > 9n$, we move all the corners c_{i+1}, \dots, c_n to the left by a distance of $\approx x(c_{i+1}) - x(c_i) - 6n$, so that the new distance is $\approx 6n$. We then do a similar truncation of vertical edges, and the resulting polyomino P'' has area $O(n^4)$. We define G^* as the induced graph $G^* := G(P'')$ and then compute a maximum matching M in G^* using a multiple-source multiple-sink maximum flow algorithm [6, 13]. The structural lemmas are used to ensure that the number of uncovered cells in maximum domino packings of the original polyomino P and the reduced P'' are the same, so it follows that a maximum domino packing in P has size $|M| + \frac{\text{area}(P) - \text{area}(P'')}{2}$.

The algorithm **fast-packer** works by reducing the packing problem to finding a maximum matching in a bipartite planar graph G^* with $O(n^3)$ vertices. The number of dominos in a maximum packing in the original polyomino P is then $|M| + \frac{\text{area}(P) - |V(G^*)|}{2}$, where M is a maximum matching in G^* and $V(G^*)$ is the set of vertices of G^* . The construction of G^* requires many techniques and technical lemmas regarding the particular way we define intermediate polyominoes and graphs that are used to eventually arrive at G^* . The process

consists of five steps, and they are illustrated and described informally in Figures 2–4. We refer the reader to the full version for a detailed description of the steps and proofs that the algorithm works as claimed.

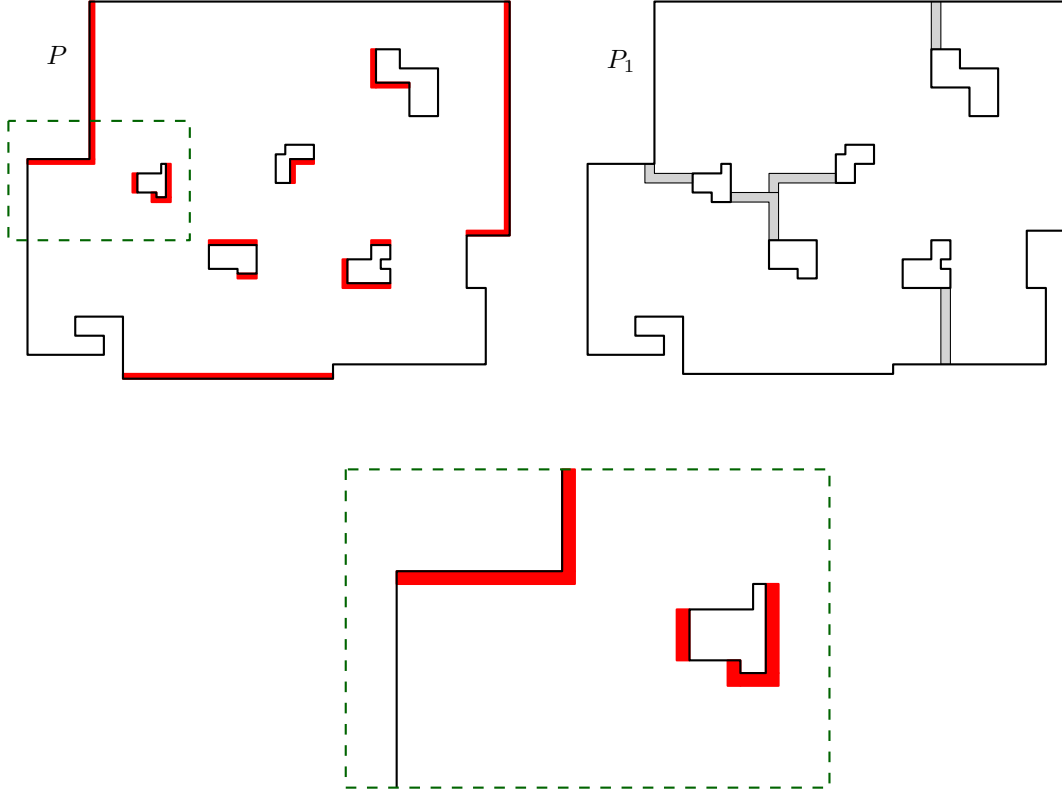
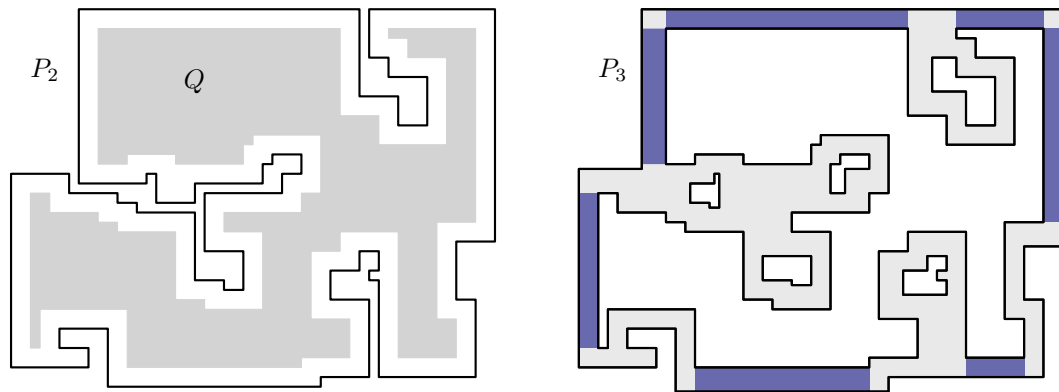


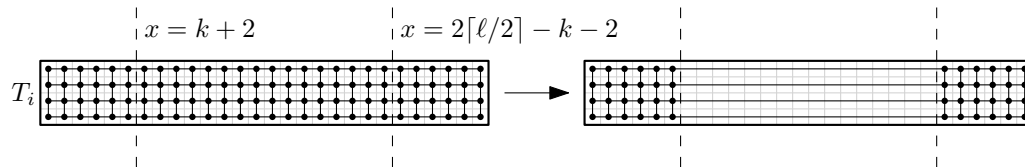
Figure 2 Steps 1 and 2 made by the **fast-packer** algorithm. Top left: In step 1, we define $P_1 \subseteq P$ to be the maximum subpolyomino with all corner coordinates even. The removed part $P \setminus P_1$ is shown in red along the edges. Top right: In step 2, we carve channels from the holes of P_1 to the outer boundary, to get a hole-free polyomino P_2 . Bottom: Closeup of the region in the dashed rectangle.

2 Preliminaries

We define a *cell* to be a 1×1 square of the form $[i, i + 1] \times [j, j + 1]$, $i, j \in \mathbb{Z}$. A subset $P \subseteq \mathbb{R}^2$ is called a *polyomino* if it is a finite union of cells. For a polyomino P , we define $G(P)$ to be the graph which has the cells in P as vertices and an edge between two cells if they share a (geometric) edge. We say that P is *connected* if $G(P)$ is a connected graph. Figure 5 (a) illustrates a connected polyomino. For a simple closed curve $\gamma \subseteq \mathbb{R}^2$, we denote by $\text{Int } \gamma$ the interior of γ . An alternative way to represent a connected polyomino is by a sequence of simple closed curves $(\gamma_0, \gamma_1, \dots, \gamma_h)$ such that (1) each of the curves follows the horizontal and vertical lines of the integral grid \mathbb{Z}^2 , (2) for each $i \in \{1, \dots, h\}$, $\text{Int } \gamma_i \subseteq \text{Int } \gamma_0$, (3) for each distinct $i, j \in \{1, \dots, h\}$, $\text{Int } \gamma_i \cap \text{Int } \gamma_j = \emptyset$, and (4) for distinct $i, j \in \{0, \dots, h\}$, $\gamma_i \cap \gamma_j \subseteq \mathbb{Z}^2$. For a connected polyomino P , there exists a unique such sequence (up to permutations of $\gamma_1, \dots, \gamma_h$) with $P = \overline{\text{Int } \gamma_0} \setminus (\bigcup_{i=1}^h \text{Int } \gamma_i)$. It is standard to reduce our tiling and packing problems to corresponding tiling and packing problems for connected polyominoes, so for simplicity we will assume that the input polyominoes to our algorithms

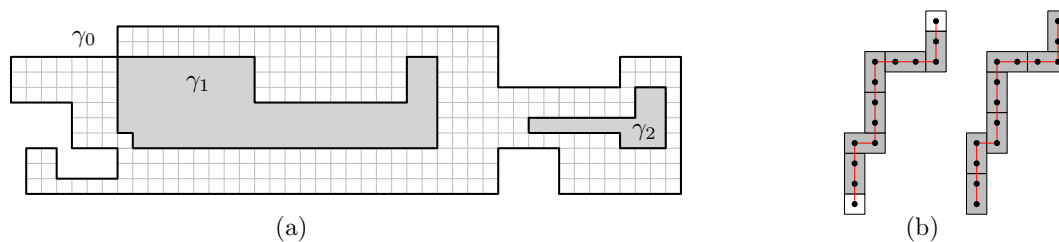


■ **Figure 3** Steps 3 and 4 made by the **fast-packer** algorithm, performed on the instance from Figure 2. Left: In step 3, we compute the grey region Q , which is obtained by offsetting the boundary of the hole-free polyomino P_2 inwards by a distance of $\Theta(n)$. In this example, Q is connected, but that is in general not the case. Right: In step 4, we consider the polyomino $P_3 := P \setminus Q$ in which all cells have distance $O(n)$ to the boundary. We identify long pipes (the dark, blue rectangles). These may be exponentially long and must therefore be reduced.



■ **Figure 4** In step 5, the algorithm **fast-packer** reduces each pipe T_i of P_3 , here of size $\ell \times k$, to a graph of polynomial size. The part of the induced graph $G(T_i)$ in between the dashed vertical lines is replaced by long horizontal edges.

are connected. The *corners* of a polyomino P (specified by a sequence $(\gamma_0, \gamma_1, \dots, \gamma_h)$), are the corners of the curves $\gamma_0, \dots, \gamma_h$. We assume that an input polyomino with n corners is represented using $O(n)$ words of memory by describing the corners of each of the curves $\gamma_0, \dots, \gamma_h$ in cyclic order.



■ **Figure 5** (a) A polyomino with two holes. (b) Extending a domino packing using an augmenting path in $G(P)$.

In this paper we will exclusively work with the L_∞ -norm when measuring distances. For two points $a, b \in \mathbb{R}^2$ we define $\text{dist}(a, b) = \|a - b\|_\infty$. For two subsets $A, B \subseteq \mathbb{R}^2$ we define

$$\text{dist}(A, B) = \inf_{(a,b) \in A \times B} \text{dist}(a, b).$$

In our analysis, A and B will always be closed and bounded (they will in fact be polyomios), and then the inf can be replaced by a min. Finally, we need the notion of the *offset* $B(A, r)$ of a set $A \subseteq \mathbb{R}^2$ by a value $r \in \mathbb{R}$. If $r \geq 0$, we define

$$B(A, r) := \{x \in \mathbb{R}^2 \mid \text{dist}(x, A) \leq r\},$$

and otherwise, we define $B(A, r) := B(A^c, -r)^c$. Note that if $r \geq 0$, we have $A \subseteq B(A, r)$ and otherwise, we have $B(A, r) \subseteq A$.

Note that a domino packing of P naturally corresponds to a matching of $G(P)$ and we will often take this viewpoint. We therefore require some basic matching terminology and a result on how to extend matchings. Let G be a graph and M a matching of G . A path (v_1, \dots, v_{2k}) of G is said to be an *augmenting path* if v_1 and v_{2k} are unmatched in M and for each $1 \leq i \leq k - 1$, v_{2i} and v_{2i+1} are matched to each other in M . Modifying M restricted to $\{v_1, \dots, v_{2k}\}$ by instead matching (v_{2i-1}, v_{2i}) for $1 \leq i \leq k$, we obtain a larger matching which now includes the two vertices v_1 and v_{2k} . See Figure 5 (b) for an illustration in the context of domino packings. We require the following basic result by Berge which guarantees that any non-maximum matching of G can always be extended to a larger matching using an augmenting path as above.

► **Lemma 1** (Berge). *Let G be a graph and M a matching of G which is not maximum. Then there exists an augmenting path between two unmatched vertices G .*

3 Tiling with squares

3.1 Naive algorithm

The naive algorithm to decide if P can be tiled with $k \times k$ tiles works as follows. Consider any convex corner c of P . A $k \times k$ square S must be placed with a corner at c . If S is not contained in P , we conclude that P cannot be tiled with $k \times k$ squares. Otherwise, we recurse on the uncovered part $P \setminus S$. When nothing is left, we conclude that P can be tiled. This algorithm runs in time polynomial in the area of P and also shows that if P can be tiled, there is a unique way to do it.

3.2 Sweep-line algorithm

For the ease of presentation, we focus on the case of deciding tileability using 2×2 squares. It is straightforward to adapt the algorithm to decide tileability by $k \times k$ squares for any fixed $k \in \mathbb{N}$, as explained in the full version.

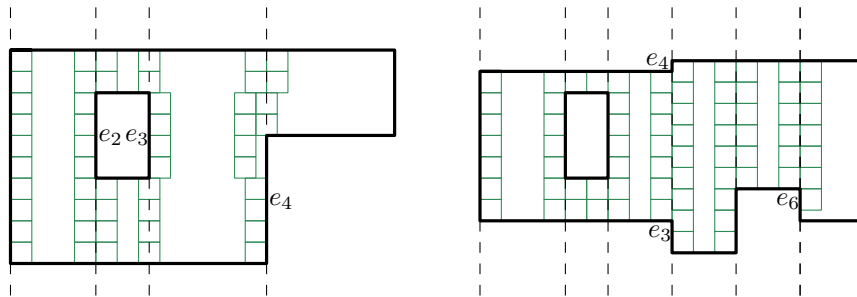
Our algorithm for deciding if a given polyomino P can be tiled with 2×2 squares uses a vertical sweep line that sweeps over P from left to right. The intuition is that the algorithm keeps track of how the tiling looks in the region of P to the left of ℓ if a tiling exists. As ℓ sweeps over P , we keep track of how the tiling pattern changes under ℓ . Each vertical edge of P that ℓ sweeps over causes changes to the tiling, and we must update our data structures accordingly.

Recall that if P is tileable, then the tiling is unique. We define $T(P) \subseteq P$ to be the union of the boundaries of the tiles in the tiling of P , i.e., such that $P \setminus T(P)$ is a set of open 2×2 squares. If P is not tileable, we define $T(P) := \perp$.

Consider the situation where the sweep line is some vertical line ℓ with integral x -coordinate $x(\ell)$. The algorithm stores a set \mathcal{I} of pairwise interior-disjoint closed intervals $\mathcal{I} = I_1, \dots, I_m \subseteq \mathbb{R}$, ordered from bottom to top. Each interval I_i has endpoints at integers

and represents the segment $I'_i := \{x(\ell)\} \times I_i$ on ℓ . In the simple case that no vertical edge of P has x -coordinate $x(\ell)$ (so that no change to the set $P \cap \ell$ happens at this point), the intervals \mathcal{I} together represent the part of ℓ in P , i.e., we have $P \cap \ell = \bigcup_{i \in [m]} I'_i$. If one or more vertical edges of P have x -coordinate $x(\ell)$, then $P \cap \ell$ changes at this point and the intervals \mathcal{I} must be updated accordingly.

For each interval I_i we store a *parity* $p(I_i) \in \{0, 1\}$, which encodes how the tiling must be at I'_i if P is tileable. To make this precise, we state the following *parity invariant* of the algorithm under the assumption that P is tileable; see also Figure 6.



■ **Figure 6** Two instances that cannot be tiled. Left: The edge e_2 splits the only interval in \mathcal{I} into two smaller intervals. Then e_3 introduces a new interval with a different parity than the existing two. The edge e_4 makes the algorithm conclude that P cannot be tiled since e_4 overlaps an interval with the wrong parity. Right: The edges e_3 and e_4 introduce new intervals that are merged with the existing one. Edge e_6 introduces an interval which is merged with the existing interval and the result has odd length, so the algorithm concludes that P cannot be tiled.

- If $p(I_i)$ and $x(\ell)$ have the same parity, then $I'_i \subseteq T(P)$, i.e., I'_i follows the boundaries of some tiles and does not pass through the middle of any tile.
- Otherwise, $I'_i \cap T(P)$ consists of isolated points, i.e., I'_i passes through the middle of some of the tiles and does not follow the boundary of any tile.

We say that two neighboring intervals I_i, I_{i+1} of \mathcal{I} are *true neighbors* if I_i and I_{i+1} share an endpoint. In addition to the parity invariant, we require \mathcal{I} to satisfy the following *neighbor invariant*: Any pair of true neighbors of \mathcal{I} have different parity.

The pseudocode of the algorithm is shown in Algorithm 1. Initially, we sort all vertical edges after their x -coordinates and break ties arbitrarily. We then run through the edges in this order. Each edge makes a change to the set $P \cap \ell$, and we need to update the intervals \mathcal{I} accordingly so that the parity and the neighbor invariants are satisfied after each edge has been handled. Figure 6 shows examples of the two cases where the algorithm concludes that there is not tiling.

Using the parity and neighbour invariants, it is proven in the full version that the algorithm returns “tileable” if and only if P is tileable. Moreover, it is shown that the algorithm can be implemented to run in $O(n \log n)$ time.

4 Simple domino packing algorithm

In this section we will present our polynomial-time algorithm `simple-packer` for finding the maximum number of 1×2 dominos that can be packed in a polyomino P . We assume that the dominos must be placed with axis-parallel edges, but they can be rotated by 90° . In any such packing, we can assume the pieces to have integral coordinates: if they do not, we

■ **Algorithm 1** Our simple sweep line algorithm for deciding if a polyomino (that may have wholes) can be tiled with 2×2 square polyominoes.

```

1 Let  $e_1, \dots, e_k$  be the vertical edges of  $P$  in sorted order.
2 for  $j = 1, \dots, k$  do
3   Let  $[y_0, y_1]$  be the interval of  $y$ -coordinates of  $e_j$ .
4   if the interior of  $P$  is to the left of  $e_j$ 
5     for each  $I_i \in \mathcal{I}$  that overlaps  $[y_0, y_1]$  do
6       if  $I_i$  and  $x(e_j)$  have different parity
7         return "no tiling"
8       Remove  $I_i$  from  $\mathcal{I}$ , let  $J := I_i \setminus [y_0, y_1]$ , and if  $J \neq \emptyset$ , add the interval(s) in
9          $J$  to  $\mathcal{I}$ .
9   else
10    Make a new interval  $I := [y_0, y_1]$  with the parity  $p(I) := x(e_j) \bmod 2$  and add
11     $I$  to  $\mathcal{I}$ .
12    if  $I$  has one or two true neighbors in  $\mathcal{I}$  that also have the same parity as  $I$ 
13      Merge those intervals in  $\mathcal{I}$ .
13  if  $j < k$  and  $x(e_{j+1}) > x(e_j)$  and some  $I_i \in \mathcal{I}$  has odd length
14    return "no tiling"
15 return "tileable"

```

can translate the pieces as far down and to the left as possible, and the corners will arrive at positions with integral coordinates. We first describe a naive algorithm which runs in polynomial time in the area of the polyomino.

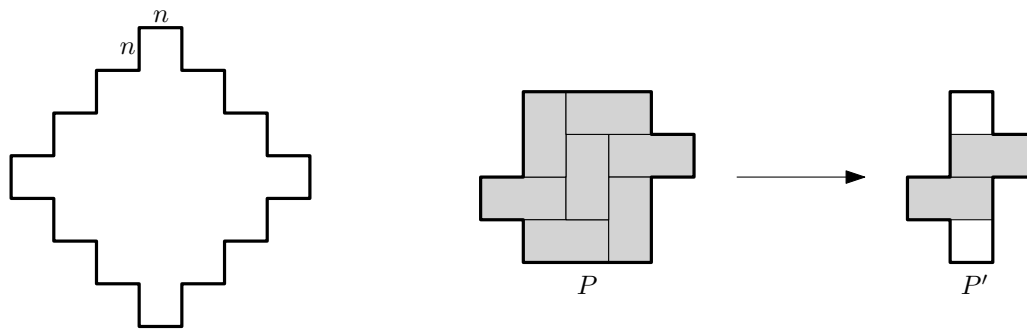
4.1 Naive algorithm

The naive algorithm considers the graph $G(P) = (V, E)$ where V is the set of cells of P and $e = (u, v) \in E$ if and only if the two cells u and v have a (geometrical) edge in common. The maximum number of 1×2 dominos that can be packed in P is exactly the size of a maximum matching of G and it is well known that such a maximum matching can be found in polynomial time in $|V|$, i.e., in the area of P .

4.2 Simple polynomial-time algorithm

Our polynomial-time algorithm, **simple-packer**, first sorts the corners of P by x -coordinates and consider the corners in this order c_1, \dots, c_n . When $x(c_{i+1}) - x(c_i) > 9n$, we move all the corners c_{i+1}, \dots, c_n to the left by a distance of $2 \lfloor \frac{x(c_{i+1}) - x(c_i)}{2} \rfloor - 6n$. We call this operation a *contraction*. The result after all of the contractions is a polyomino P' with the parities of the x -coordinates unchanged and with the difference between the x -coordinates of any two consecutive corners at most $6n$. We then consider the corners in order according to y -coordinates and do a similar truncation of the long vertical edges. We have now reduced the container P to an orthogonal polygon P'' of area at most $O(n^4)$, since the span of the x -coordinates is $O(n^2)$, as is the span of the y -coordinates. We proceed by finding maximum or perfect matchings in $G(P'')$, as described in the introduction.

For some containers P , the graph $G(P'')$ really has $\Omega(n^4)$ vertices, so **simple-packer** is indeed slower than **fast-packer**. For instance when the boundary of P consists of four "staircases", each consisting of $n/4$ vertices, where each step has width and height n ; see Figure 7 (left). Here, **fast-packer** will remove most of the interior, leaving a layer of cells of thickness $O(n)$ around the boundary, but **simple-packer** will not make any contractions.



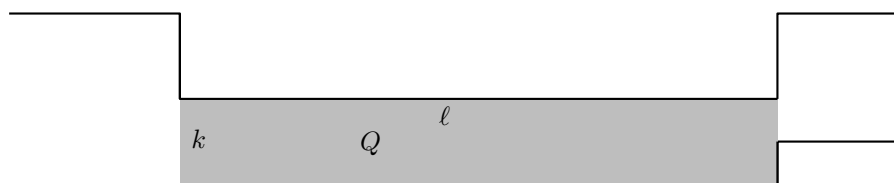
■ **Figure 7** Left: A polyomino with area $\Omega(n^4)$ that `simple-packer` will not reduce. Right: If we truncate edges so that consecutive x -coordinates have difference either 1 or 2 (keeping the parities invariant), then there may be more uncovered cells in a maximum packing of the reduced instance than in the original.

One might be tempted to think that we can even truncate the edges so that the difference between consecutive x - and y -coordinates is either 1 or 2, keeping the parity of all coordinates. However, this does not work, as seen in Figure 7 (right). Two dominos can be packed in the reduced container P' , and the reduction decreases the area by eighth cells, so the formula would give that the original container P has room for six dominos, but there is actually room for seven.

4.3 Structural results on polyominoes and domino packings

Building up to our structural results on domino packings, we require a few definition and simple lemmas. We first introduce the notion of a *pipe* (see Figure 8) and *consistent parity*.

► **Definition 2.** Let P and Q be polyominoes with $Q \subseteq P$. We say that Q is a pipe of P if Q is rectangular and both vertical edges of Q or both horizontal edges of Q are contained in edges of P . The width of the pipe is the distance between this pair of edges. The length of the pipe is the distance between the other pair of edges. We say that a pipe is long if its length is at least 3 times its width.

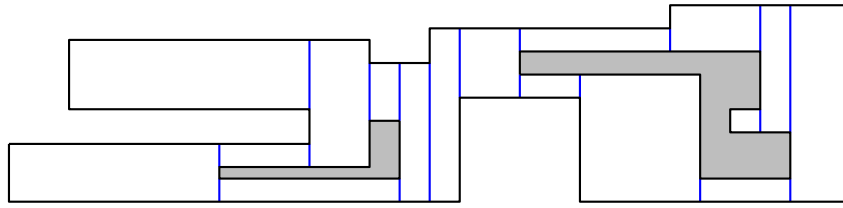


■ **Figure 8** A pipe Q of width k and length ℓ .

► **Definition 3.** We say that a polyomino P has consistent parity if all first coordinates of the corners of P have the same parity and likewise for the second coordinates. Equivalently, P has consistent parity if there exists an open 2×2 square, S , such that for all choices of integers i, j and $S' = S + (2i, 2j)$, either $S' \subseteq P$ or $S' \cap P = \emptyset$.

Variations of the following lemma are well-known. We present a proof for completeness.

► **Lemma 4.** Let P be an orthogonal polygon with n corners and h holes. P can be divided into at most $n/2 + h - 1$ rectangular pieces by adding only vertical line segments to the interior of P . If P is a polyomino, the rectangular pieces can be chosen to be polyominoes too.



■ **Figure 9** A partition of a polyomino with two holes into rectangles using vertical line segments (blue).

Proof. For each concave corner of the polygon we add a vertical line segment in the interior of the polygon starting from that corner and going upwards or downwards (depending on the rotation of the given corner). This is illustrated in Figure 9. Let s be the number of line segments added. It is easy to check that this gives a partition of P into exactly $s - h + 1$ rectangles. With h holes, the number of concave corners is $n/2 + 2(h - 1)$, so also $s \leq n/2 + 2(h - 1)$ and the result follows. ◀

Note that for a polygon with n corners, $h \leq (n - 4)/4$, so we have the following trivial corollary.

► **Corollary 5.** *The number of rectangular pieces in Lemma 4 is at most $\frac{3}{4}n - 2$.*

We next show that the property of consisting parity is preserved under integral offsets.

► **Lemma 6.** *Let P be a polyomino. If P has consistent parity, then $B(P, 1)$ and $B(P, -1)$ have consistent parity.*

Proof. Suppose P has consistent parity. Let S be a 2×2 square as in Definition 3. Define $S_1 = S + (1, 1)$. It is easy to check that for all choices of integers i, j and $S'_1 := S_1 + (2i, 2j)$, either $S'_1 \subseteq B(P, 1)$ or $S'_1 \cap B(P, 1) = \emptyset$. Thus $B(P, 1)$ has consistent parity. The argument that $B(P, -1)$ has consistent parity is similar. ◀

► **Lemma 7.** *Let P be a connected polyomino of consistent parity and without holes. Define $L_1 = B(P, 1) \setminus P$ and $L_{-1} = P \setminus B(P, -1)$. Then $G(L_1)$ and $G(L_{-1})$ both have a Hamiltonian cycle of even length.*

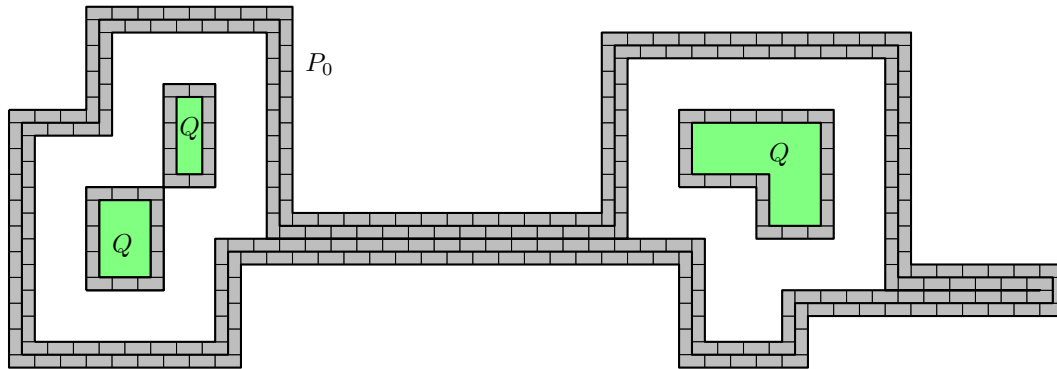
Proof. To obtain a Hamiltonian cycle of $G(L_1)$, we can simply trace P around the outside of its boundary, visiting all cells of L_1 in a cyclic order. The corresponding closed trail of $G(L_1)$ visits each vertex at least once. The assumption of consistent parity is easily seen to imply that we in fact visit each vertex exactly once, so the obtained trail is a Hamiltonian cycle. The graph $G(L_1)$ is bipartite, so the cycle has even length. The argument that $G(L_{-1})$ has a Hamiltonian cycle of even length is similar. ◀

With the above in hand, we are ready to state and prove our main structural results on domino packings. They are presented in Lemma 8 and Lemma 10.

► **Lemma 8.** *Let P and P_0 be polyominoes such that $P_0 \subseteq P$, P_0 has no holes, and P_0 has consistent parity. Let the total number of corners of P and P_0 be n . Define $r = \lfloor \frac{3}{8}n \rfloor$ and $Q = B(P_0, -r)$. There exists a maximum packing of P with 1×2 dominos which restricts to a tiling of Q .*

Let us briefly pause to explain the importance of Lemma 8. Suppose that P contains a region Q as described. Then Lemma 8 tells us that *any* domino tiling of Q can be extended to a maximum domino packing of P . We can thus disregard Q and focus on finding a maximum packing of $P \setminus Q$, thus reducing the problem to a smaller instance. This is one of our key tools for reducing the size of the original polyomino P to a matching problem of polynomial size.

Proof. It follows from Lemma 6 that Q has consistent parity, and it can thus be tiled with 2×2 squares and hence with dominos. Let \mathcal{Q} be a tiling of Q .



■ **Figure 10** The polyomino P_0 and the offset Q (shown in green). The figure also illustrates the “layers” A_i and their domino tilings, \mathcal{A}_i .

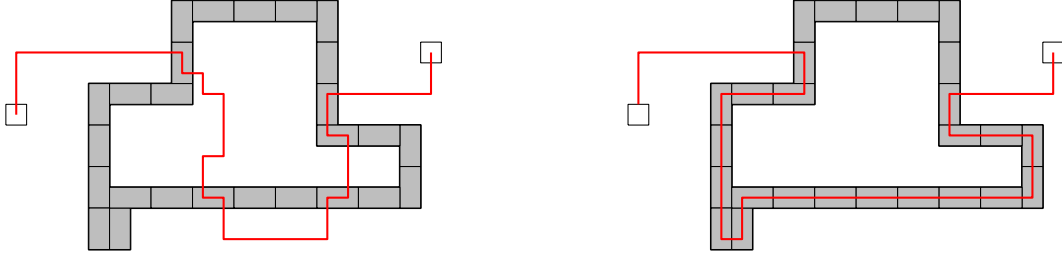
Define $R = P \setminus P_0$ and note that R has at most n corners. It follows from Corollary 5 that R can be partitioned into less than $\frac{3}{4}n$ rectangular polyominoes. Each of these rectangles has a domino packing with at most one uncovered cell (which happens when the total number of cells in the rectangle is odd). Fix such a packing \mathcal{R} of the rectangles of R with dominos.

We next describe a tiling of $P_0 \setminus Q$ as follows. For integers $1 \leq i \leq r$ we define, $A_i = B(P_0, -i + 1) \setminus B(P_0, -i)$. Intuitively, we can construct Q from P_0 by peeling off the “layers” A_i of P_0 one at a time. Let $i \in \{1, \dots, r\}$ be fixed. As P_0 has consistent parity, it follows from Lemma 6 that $B(P_0, -i + 1)$ has consistent parity. It is also easy to check that $B(P_0, -i + 1)$ has no holes either, and it then follows from Lemma 7 that each connected component of $G(A_i)$ has a Hamiltonian cycle of even length. These cycles give rise to a natural tiling of A_i ; if (v_1, \dots, v_{2k}) is the sequence of cells corresponding to such a cycle, then $\{v_1 \cup v_2, v_3 \cup v_4, \dots, v_{2k-1} \cup v_{2k}\}$ is a tiling of the cells of the cycle, and the union of such tilings over all connected components in $G(A_i)$ gives a tiling of A_i with dominos. Denote this tiling by \mathcal{A}_i . See Figure 10 for an illustration of this construction.

Combining the tilings $\mathcal{A}_1, \dots, \mathcal{A}_r$ and \mathcal{Q} with the packing \mathcal{R} , we obtain a domino packing, \mathcal{P} , of P where at most $\frac{3}{4}n$ cells of P are uncovered. We now wish to extend this packing to a maximum packing in a way where we do not alter the tiling \mathcal{Q} of Q . If we can do this, the result will follow. Let M be the matching corresponding to \mathcal{P} in $G(P)$. We make the following claim.

▷ **Claim 9.** Let $k \leq r$. Suppose that the matching M can be extended to a matching of size $|M| + k$. Then this extension can be made using a sequence C_1, \dots, C_k of k augmenting paths one after the other (that is, C_i is an augmenting path *after* the matching has been extended using C_1, \dots, C_{i-1}) such that for each $i \in \{1, \dots, k\}$, we have that C_i only uses vertices of $G(R \cup \bigcup_{j=1}^i A_j)$.

Before proving this claim, we first argue how the result follows. Since there are less than $\frac{3}{4}n$ unmatched vertices in M , we can extend M to a maximum matching using at most $r = \lfloor \frac{3}{8}n \rfloor$ augmenting paths. By the claim, these paths can be chosen so that they avoid the vertices of $G(Q)$. In particular, we never alter the matching of $G(Q)$, so the final maximum matching restricted to $G(Q)$ is just the tiling \mathcal{Q} .

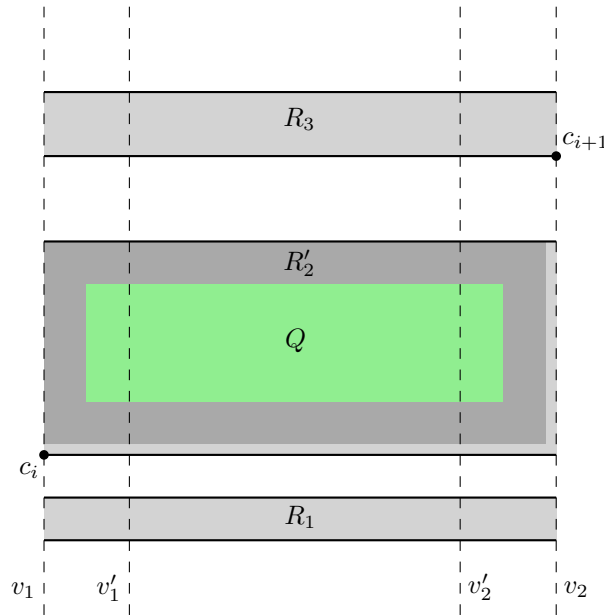


■ **Figure 11** Left: An alternating path between two unmatched vertices which enters a connected component of $G(A_k)$. Right: Modifying the alternating path using the the Hamiltonian cycle of the connected component.

We proceed to prove the claim by induction on k . The statement is trivial for $k = 0$, so let $1 \leq k \leq r$ satisfy the assumptions of the claim and suppose inductively that C_1, \dots, C_{k-1} can be chosen such that for each $i \in \{1, \dots, k-1\}$, we have that C_i only uses vertices of $G(R \cup \bigcup_{j=1}^i A_j)$. After augmenting the matching using C_1, \dots, C_{k-1} , we have only modified the matching restricted to $G(R \cup \bigcup_{j=1}^k A_j)$. By Lemma 1, we can find an augmenting path C'_k connecting two unmatched vertices u, v of $G(P)$. We will modify C'_k to a path C_k with $C_k \subseteq R \cup \bigcup_{j=1}^k A_j$. Write $C'_k : u = u_1, u_2, \dots, u_{2\ell} = v$. Let D be a Hamiltonian cycle of one of the connected components of $G(A_k)$; see Figure 11. If the path C'_k ever enters the vertices of D , we let i be minimal such that $u_i \in D$ and j be maximal such that $u_j \in D$. We can now replace the subpath u_i, u_{i+1}, \dots, u_j of C'_k with part of the Hamiltonian cycle D . Whether we go clockwise or counterclockwise along D depends on whether u_i is matched with u_{i+1} in a clockwise or counterclockwise fashion in D . We do the same modification for every Hamiltonian cycle D corresponding to a connected component of $G(A_k)$ that C'_k intersects. Note that each cycle D partitions the vertices $G(P) \setminus D$ into an interior and an exterior part. Since P_0 has no holes and $u, v \in R$, the original path C'_k enters D from the exterior at u_i and likewise leaves D into the exterior at u_j . Also note that Q is contained in the interior parts of the cycles of $G(A_k)$. It then follows that the final resulting path C_k avoids Q and A_j for $j > k$, so it is contained in $R \cup \bigcup_{j=1}^k A_j$. ◀

Lemma 8 allows us to “ignore” parts of the polyomino P with distance $\Omega(n)$ to the boundary. In order to argue that the answer output by `simple-packer` is correct, we also need to argue that we can ignore long pipes (see Definition 2). This is what motivates the following lemma which intuitively yields a reduction for shortening long pipes. We defer the proof to the full version.

► **Lemma 10.** *Let $k, \ell \in \mathbb{N}$ with ℓ even. Let $L \subseteq [-1, 0] \times [0, k]$, $R \subseteq [\ell, \ell + 1] \times [0, k]$ be polyominoes and define $P = L \cup R \cup ([0, \ell] \times [0, k])$. Color the cells of the plane in a chessboard like fashion and let b and w be respectively the number of black and white cells contained in P . Assume without loss of generality that $b \geq w$. If $\ell \geq 2k$, then the number of uncovered cells in a maximum domino packing of P is exactly $b - w$. Moreover, there exists a maximum domino packing such that the rectangle $[k + 1, \ell - k - 1] \times [0, k]$ is completely covered and all dominos intersecting the rectangle are horizontal.*



■ **Figure 12** A contraction made by the algorithm `simple-packer` with one fat and two skinny rectangles. The algorithm moves all corners c_{i+1}, \dots, c_n to the left, essentially contracting the area between the vertical lines v'_1 and v'_2 to nothing.

4.4 Correctness of the algorithm

We now verify that the number of uncovered cells in maximum packings is invariant under a single contraction, and the correctness of the algorithm hence follows. To this end, suppose that $x(c_{i+1}) - x(c_i) > 9n$, so that we move the corners c_{i+1}, \dots, c_n to the left; see Figure 12. It is clear that a domino packing with exactly ℓ uncovered cells after the contraction gives rise to a domino packing with exactly ℓ uncovered cells before the contraction, simply by inserting extra horizontal dominos in the rectangles that were contracted. For the converse, let v_1 and v_2 be vertical lines with x -coordinates $x(c_i)$ and $x(c_{i+1})$, respectively, and let V be the vertical strip bounded by v_1 and v_2 . The intersection $P \cap V$ is a collection of disjoint rectangles R_1, \dots, R_k of width $x(c_{i+1}) - x(c_i)$ and various heights. We define a rectangle R_i to be *fat* if its height is more than $3n$, and otherwise R_i is *skinny*. We now define a polyomino P_0 in order to apply Lemma 8. For each fat rectangle R_i , we let $R'_i \subseteq R_i$ be the maximum rectangle with even coordinates and add R'_i to P_0 . As each rectangle R_i corresponds to exactly two horizontal edges, the number of rectangles k is at most $n/4$ and in particular, the number of corners of $P \setminus P_0$ is at most $2n$. Letting $Q := B(P_0, -\lfloor 3n/2 \rfloor)$, we get from Lemma 8 that there exists a maximum packing of P that restricted to Q is a tiling.

We define $P_1 := P \setminus Q$ and observe that the contraction corresponds to contracting a set of long pipes in P_1 . These pipes are the skinny rectangles R_i and the parts of the fat rectangles vertically above and below the removed part Q . We therefore get from Lemma 10 that a maximum packing before the contraction having ℓ uncovered cells, gives rise to a packing of the contracted polyomino with exactly ℓ uncovered cells.

References

- 1 Danièle Beauquier and Maurice Nivat. On translating one polyomino to tile the plane. *Discrete & Computational Geometry*, 6:575–592, 1991. doi:10.1007/BF02574705.
- 2 Danièle Beauquier, Maurice Nivat, Eric Remila, and Mike Robson. Tiling figures of the plane with two bars. *Computational Geometry*, 5(1):1–25, 1995. doi:10.1016/0925-7721(94)00015-N.
- 3 Robert Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 1(66), 1966. doi:10.1090/memo/0066.
- 4 Fran Berman, David Johnson, Tom Leighton, Peter W. Shor, and Larry Snyder. Generalized planar matching. *Journal of Algorithms*, 11(2):153–184, 1990. doi:10.1016/0196-6774(90)90001-U.
- 5 Francine Berman, Frank Thomson Leighton, and Lawrence Snyder. Optimal tile salvage, 1982. Technical report, Purdue University, Department of Computer Sciences, <https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1321&context=cstech>.
- 6 Glencora Borradaile, Philip N. Klein, Shay Mozes, Yahav Nussbaum, and Christian Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. *SIAM Journal on Computing*, 46(4):1280–1303, 2017. doi:10.1137/15M1042929.
- 7 Chen-Fu Chien, Shao-Chung Hsu, and Jing-Feng Deng. A cutting algorithm for optimizing the wafer exposure pattern. *IEEE Transactions on Semiconductor Manufacturing*, 14(2):157–162, 2001.
- 8 J.H Conway and J.C Lagarias. Tiling with polyominoes and combinatorial group theory. *Journal of Combinatorial Theory, Series A*, 53(2):183–208, 1990. doi:10.1016/0097-3165(90)90057-4.
- 9 Dirk K. de Vries. Investigation of gross die per wafer formulas. *IEEE Transactions on Semiconductor Manufacturing*, 18(1):136–139, 2005.
- 10 Dania El-Khechen, Muriel Dulieu, John Iacono, and Nikolaj Van Omme. Packing 2×2 unit squares into grid polygons is NP-complete. In *Proceedings of the 21st Canadian Conference on Computational Geometry (CCCG 2009)*, pages 33–36, 2009.
- 11 Robert J. Fowler, Michael S. Paterson, and Steven L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Information processing letters*, 12(3):133–137, 1981.
- 12 George Gamow and Marvin Stern. *Puzzle-math*. Macmillan, 1958.
- 13 Pawel Gawrychowski and Adam Karczmarz. Improved bounds for shortest paths in dense distance graphs. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, pages 61:1–61:15, 2018. doi:10.4230/LIPIcs.ICALP.2018.61.
- 14 S. W. Golomb. Checker boards and polyominoes. *The American Mathematical Monthly*, 61(10):675–682, 1954. doi:10.1080/00029890.1954.11988548.
- 15 Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136, 1985.
- 16 Takashi Horiyama, Takehiro Ito, Keita Nakatsuka, Akira Suzuki, and Ryuhei Uehara. Complexity of tiling a polygon with trominoes or bars. *Discret. Comput. Geom.*, 58(3):686–704, 2017. doi:10.1007/s00454-017-9884-9.
- 17 S. Jang, J. Kim, T. Kim, H. Lee, and S. Ko. A wafer map yield prediction based on machine learning for productivity enhancement. *IEEE Transactions on Semiconductor Manufacturing*, 32(4):400–407, 2019.
- 18 C. Kenyon and R. Kenyon. Tiling a polygon with rectangles. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science (FOCS 1992)*, pages 610–619, 1992.
- 19 Hanno Melzner and Alexander Olbrich. Maximization of good chips per wafer by optimization of memory redundancy. *IEEE Transactions on Semiconductor Manufacturing*, 20(2):68–76, 2007.
- 20 Gary L. Miller and Joseph Naor. Flow in planar graphs with multiple sources and sinks. *SIAM Journal on Computing*, 24(5):1002–1017, 1995. doi:10.1137/S0097539789162997.

- 21 Shay Mozes and Christian Wulff-Nilsen. Shortest paths in planar graphs with real lengths in $O(n \log^2 n / \log \log n)$ time. In *16th Annual European Symposium on Algorithms (ESA 2010)*, 2010. doi:10.1007/978-3-642-15781-3_18.
- 22 Igor Pak, Adam Sheffer, and Martin Tassy. Fast domino tileability. *Discrete & Computational Geometry*, 56(2):377–394, 2016. doi:10.1007/s00454-016-9807-1.
- 23 Igor Pak and Jed Yang. Tiling simply connected regions with rectangles. *Journal of Combinatorial Theory, Series A*, 120(7):1804–1816, 2013. doi:10.1016/j.jcta.2013.06.008.
- 24 Eric Rémila. Tiling a polygon with two kinds of rectangles. *Discrete Comput. Geom.*, 34(2):313–330, 2005. doi:10.1007/s00454-005-1173-3.
- 25 William P. Thurston. Conway’s tiling groups. *The American Mathematical Monthly*, 97(8):757–773, 1990.
- 26 H.A.G. Wijshoff and J. van Leeuwen. Arbitrary versus periodic storage schemes and tessellations of the plane using one type of polyomino. *Information and Control*, 62(1):1–25, 1984. doi:10.1016/S0019-9958(84)80007-8.

On Cyclic Solutions to the Min-Max Latency Multi-Robot Patrolling Problem

Peyman Afshani ✉

Department of Computer Science,
Aarhus University, Denmark

Kevin Buchin ✉ 

Department of Computer Science,
TU Dortmund, Germany

Maarten Löffler ✉

Department of Information and Computing Sciences,
Utrecht University, The Netherlands

Benjamin Raichel ✉

Department of Computer Science, University of
Texas at Dallas, Richardson, TX, USA

Haotian Wang ✉

Department of Computer Science,
Rutgers University, New Brunswick, NJ, USA

Mark de Berg ✉ 

Department of Mathematics and Computer Science,
TU Eindhoven, The Netherlands

Jie Gao ✉ 

Department of Computer Science,
Rutgers University, New Brunswick, NJ, USA

Amir Nayyeri ✉

School of Electrical Engineering and Computer
Science, Oregon State University,
Corvallis, OR, USA

Rik Sarkar ✉

School of Informatics,
University of Edinburgh, UK

Hao-Tsung Yang ✉

School of Informatics,
University of Edinburgh, UK

Abstract

We consider the following surveillance problem: Given a set P of n sites in a metric space and a set R of k robots with the same maximum speed, compute a *patrol schedule* of minimum latency for the robots. Here a patrol schedule specifies for each robot an infinite sequence of sites to visit (in the given order) and the latency L of a schedule is the maximum latency of any site, where the latency of a site s is the supremum of the lengths of the time intervals between consecutive visits to s .

When $k = 1$ the problem is equivalent to the travelling salesman problem (TSP) and thus it is NP-hard. For $k \geq 2$ (which is the version we are interested in) the problem becomes even more challenging; for example, it is not even clear if the decision version of the problem is decidable, in particular in the Euclidean case.

We have two main results. We consider *cyclic solutions* in which the set of sites must be partitioned into ℓ groups, for some $\ell \leq k$, and each group is assigned a subset of the robots that move along the travelling salesman tour of the group at equal distance from each other. Our first main result is that approximating the optimal latency of the class of cyclic solutions can be reduced to approximating the optimal travelling salesman tour on some input, with only a $1 + \varepsilon$ factor loss in the approximation factor and an $O((k/\varepsilon)^k)$ factor loss in the runtime, for any $\varepsilon > 0$. Our second main result shows that an optimal cyclic solution is a $2(1 - 1/k)$ -approximation of the overall optimal solution. Note that for $k = 2$ this implies that an optimal cyclic solution is optimal overall. We conjecture that this is true for $k \geq 3$ as well.

The results have a number of consequences. For the Euclidean version of the problem, for instance, combining our results with known results on Euclidean TSP, yields a PTAS for approximating an optimal cyclic solution, and it yields a $(2(1 - 1/k) + \varepsilon)$ -approximation of the optimal unrestricted (not necessarily cyclic) solution. If the conjecture mentioned above is true, then our algorithm is actually a PTAS for the general problem in the Euclidean setting. Similar results can be obtained by combining our results with other known TSP algorithms in non-Euclidean metrics.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Approximation, Motion Planning, Scheduling

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.2

Related Version *Full Version*: <http://arxiv.org/abs/2203.07280>



© Peyman Afshani, Mark de Berg, Kevin Buchin, Jie Gao, Maarten Löffler, Amir Nayyeri, Benjamin Raichel, Rik Sarkar, Haotian Wang, and Hao-Tsung Yang; licensed under Creative Commons License CC-BY 4.0

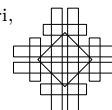
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 2; pp. 2:1–2:14

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Funding *Mark de Berg*: Supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.

Jie Gao: This work is supported by NSF OAC-1939459, CCF-2118953 and CCF-1934924.

Benjamin Raichel: Partially supported by NSF CAREER Award 1750780.

1 Introduction

We study the following problem, motivated by the problem of monitoring a fixed set of locations using autonomous robots: We are given a set $P = \{s_1, \dots, s_n\}$ of n sites in a metric space as well as a set $R = \{r_1, \dots, r_k\}$ of k robots. We assume the robots have the same maximum speed, called the *unit speed*, and their task is to repeatedly visit (i.e., survey) the sites such that the maximum time during which any site is left unmonitored is minimized. More precisely, we wish to compute a *patrol schedule*; that is, an infinite sequence of sites to visit for each robot, of minimum *latency*. Here the latency of a site s_i is the supremum of the length of the time intervals between consecutive visits of s_i , and the latency of the patrol schedule is the maximum latency over all the sites.

Related Work. For $k = 1$, the problem reduces to the Traveling Salesman Problem. To see this, consider the time interval $[0, 3L]$, where L is the optimal latency, and observe that every site is visited at least twice by the robot in this time interval. Let $L' \leq L$ be the maximum length of time between two consecutive visits of a site. Then there exists a site that is visited at times t_0 and $t_0 + L'$ and all other sites are visited at least once in the time interval $(t_0, t_0 + L')$. Hence, if an optimal solution has latency L , there is a TSP tour of length at most L . The converse is clearly true as well – by repeatedly traversing a TSP tour of length L we obtain a patrol schedule of latency L – and so the TSP problem is equivalent to the patrol problem for a single robot. Since TSP is NP-hard even in the Euclidean case [16] we will focus on approximation algorithms. There are efficient approximation algorithms for TSP and, hence, for the patrolling problem for $k = 1$. In particular, there is a $(3/2)$ -approximation for metric TSP [5] (which was slightly improved very recently [10]) and a PTAS for Euclidean TSP [4, 15]. However, it seems difficult to generalize these solutions to the case $k \geq 2$, because it seems non-trivial to get a grip on the structure of optimal solutions in this case. We will mention some of the major challenges shortly.

There has been a lot of work on such surveillance problems in the robotics community [7, 9, 14, 21, 17, 18]. Most previous work, however, focused on either practical settings or aspects of the problems other than finding the best approximation factor. There are two papers that provide theoretical guarantees for the weighted version of the problem, where sites of higher weight require more frequent patrols. Alamdari et al. [2] provided a $O(\log n)$ -approximation algorithm for the weighted problem for $k = 1$. (Due to existence of weights, a TSP tour may no longer be optimal for $k = 1$.) Afshani et al. [1] studied the problem for $k \geq 1$ and they present an $O(k^2 \log \frac{w_{\max}}{w_{\min}})$ -approximation algorithm, where w_{\max} and w_{\min} are the maximum and the minimum weights of the sites.

Related Problems. As already mentioned, the TSP problem can be viewed as a special case of the problem for unweighted sites and for $k = 1$. Another related problem is the *k-path cover* problem where we want to find k paths that cover the vertices of an edge-weighted graph such that the maximum length of the paths is minimized. This problem has a 4-approximation algorithm [3]. Another problem is the problem of covering all the sites with k trees that minimize the maximum length of the trees; this problem is known as the *min-max tree cover*

problem and it has constant-factor approximation algorithms [3, 13] with $8/3$ being the current record [20]. The k -cycle cover problem is similar, except that we want to use k cycles (instead of paths or trees); again constant-factor approximation algorithms are known, with $16/3$ being the current record [20]. If the goal is to minimize the sum of all cycle lengths, there is a 2-approximation for the metric setting and a PTAS in the Euclidean setting [11, 12]. Our problem is also related to (but different from) the *vehicle routing problem* (VRP) [6], which asks for k tours, starting from a given depot, that minimize the total transportation cost under various constraints; see the surveys by Golden et al. [8] or Tóth and Vigo [19].

Our Results. All covering problems mentioned above are obviously decidable. The question of decidability for the patrolling problem seems non-trivial. However, since patrol schedules are infinite sequences and thus it is not even clear how to guess a solution¹. To tackle this issue, we consider the class of *cyclic solutions*. In a cyclic solution the set P of sites is partitioned into $\ell \leq k$ subsets P_1, \dots, P_ℓ , and each subset P_i is assigned k_i robots, where $\sum_{i=1}^{\ell} k_i = k$. The k_i robots are then distributed evenly along a TSP tour of P_i , and they traverse the tour at maximum speed. Thus, the latency of the sites in P_i equals $\|T_i\|/k_i$, where $\|T_i\|$ is the length of the TSP tour of P_i .

The significance of this definition is that in Section 3 we prove that (in any metric space) the best cyclic solution is a $2(1 - 1/k)$ -approximation of the optimal solution in terms of maximum latency. We do this by transforming an optimal solution to a cyclic one, with only a $2(1 - 1/k)$ factor loss in the approximation ratio. This proof is highly non-trivial and involves a number of graph-theoretic arguments and carefully inspecting the coordinated motion of the k robots, cutting them up at proper locations, and re-gluing the pieces together to form a cyclic solution. In combination with this, in Section 4 we prove that, given a γ -approximation algorithm for TSP, for any fixed k and $\varepsilon > 0$, we can obtain a $(1 + \varepsilon)\gamma$ -approximation of the best cyclic schedule in polynomial time. Therefore, in the Euclidean setting, we can use a known PTAS to obtain a $(1 + \varepsilon)$ -approximation to the *best cyclic* solution and in the general metric setting, we can use known approximation algorithms for TSP [10] to get a 1.5-approximation to the *best cyclic* solution. Together with the results in Section 3 these lead to a $(2 - 2/k + \varepsilon)$ -approximation algorithm for the Euclidean case, and a $(3 - 3/k)$ -approximation for general metrics.

We conjecture that the best cyclic solution is in fact the best overall solution. If this is true, then our algorithm in Section 4 already gives a PTAS in the Euclidean setting. Observe that a corollary of our result in Section 3 is that the conjecture holds for $k = 2$. We remark that there is an easy proof showing the existence of a cyclic 2-approximation solution (See Section 2.2). Our new bound $2(1 - 1/k)$ is a significant improvement when k is a small constant. For example, for $k = 3$, we get that a cyclic $4/3$ approximate solution exists, and for $k = 2$ –as mentioned above– that there is a cyclic optimal solution.

2 Challenges, Notation, and Problem Statement

2.1 Notation and Problem Statement

Let (P, d) be a metric space on a set P of n sites, where the distance between two sites $s_i, s_j \in P$ is denoted by $d(s_i, s_j)$. Following Afshani et al. [1], we model the metric space in the following way. We take the undirected complete graph $G = (P, P \times P)$, and we view each edge

¹ If we assume that all distances are integers and we want to decide whether the latency is at most a given integer ℓ , then we can guess a solution. These assumptions, however, do not hold in the Euclidean case, even if the coordinates of sites are rational.

$(s_i, s_j) \in P \times P$ as an interval (that is, a continuous 1-dimensional space) of length $d(s_i, s_j)$ in which the robot can travel. This transforms the discrete metric space (P, d) into a continuous metric space $C(P, d)$. From now on, and with a slight abuse of terminology, when we talk about the metric space (P, d) we actually mean the continuous metric space $C(P, d)$.

We allow the robots to “stay” on a site for any amount of time. This implies it never helps if a robot moves slower than the maximum speed: indeed, the robot may as well move at maximum speed towards the next site and stay a bit longer at that site. Also, it does not help to have a robot start at time $t = 0$ “in the middle” of an edge, so we can assume all robots start at some sites at the beginning. A *schedule* of a robot r_j is defined as a continuous function $f_j : \mathbb{R}^{\geq 0} \rightarrow C(P, d)$, where $f_j(t)$ specifies the position of r_j at time t . The unit-speed constraint implies that a valid schedule must satisfy $d(f_j(t_1), f_j(t_2)) \leq |t_1 - t_2|$ for all t_1, t_2 . A *schedule for the collection R of robots*, denoted by $\sigma(R)$, is a collection of schedules f_j , one for each robot $r_j \in R$. Note that we allow robots to be at the same location at the same time.

We say that a site $s_i \in P$ is *visited* at time t if $f_j(t) = s_i$ for some robot r_j . Given a schedule $\sigma(R)$, the *latency* L_i of a site s_i is defined as follows.

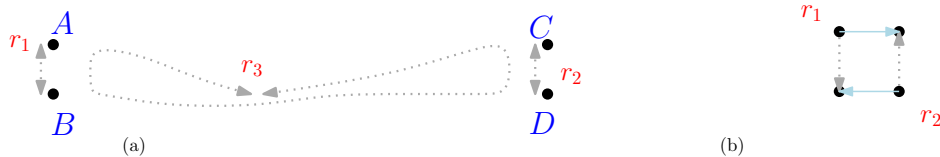
$$L_i = \sup_{0 \leq t_1 < t_2} \{|t_2 - t_1| : s_i \text{ is not visited during the time interval } (t_1, t_2)\}$$

We only consider schedules where the latency of each site is finite. Clearly such schedules exist; e.g., a robot can repeatedly traverse a TSP tour of the sites. Given a metric space (P, d) and a collection R of k robots, the *(multi-robot) patrol-scheduling problem* is to find a schedule $\sigma(R)$ minimizing the latency $L := \max_i L_i$, the maximum latency of any site.

2.2 Challenges

The problem of scheduling multiple robots is quite challenging and involves several subtleties, caused by the fact that patrol schedules are infinite sequences. For example, the time intervals between consecutive visits of any given site might increase continuously, and so we have to define the latency of a site using the notion of supremum rather than maximum. Moreover, for $k > 1$, it is not even clear if the problem is decidable: Given a set of n points in the Euclidean plane, an integer $k > 1$, and a value L , is it decidable if there exists a patrol schedule for the k robots such that the maximum latency is bounded by L ? As already mentioned, a corollary of our results in Section 3 is that for $k = 2$ there exists an optimal cyclic solution and thus for $k = 2$ the answer to the above question is yes.

A severe challenge is that, since patrol schedules are infinite sequences, it is difficult to rule out chaotic solutions where the robots visit the sites in a way that avoids any sort of repeated pattern. Indeed, optimal solutions can behave so chaotically that they require an infinite sequence of bits to describe. For instance, consider the left situation in Figure 1,



■ **Figure 1** (a) Four points A, B, C , and D form a short and wide rectangle. Robots r_1, r_2 , and r_3 can have infinite “unpredictable” optimal patrol schedules. (b) Robots r_1 and r_2 can move to the other diagonal in two different ways.

where we have three robots and four points A, B, C, D that are the vertices of a thin rectangle. To obtain the optimal latency, it suffices that r_1 moves back and forth between A and B , and r_2 moves back and forth between C and D . Since r_3 cannot be used to decrease the latency – it will take r_3 too much time to go from A, B to C, D – it can behave as chaotically as it wants, thus causing the description of the patrol schedule to be arbitrarily complicated. This is even possible using only two robots: Consider four sites that form a unit square and two robots placed on opposite corners of the square; see the right situation in Figure 1. An optimal schedule is then an infinite sequence of steps, where in each step both robots move counterclockwise or both move clockwise. Such a schedule need not be cyclic and, hence, may require an infinite sequence of bits to describe. Of course, in both cases we know optimal cyclic solutions exist, and such solutions can be described using finitely many bits. We conjecture that this should be true in general:

► **Conjecture 1.** *For the k -robot patrolling problem with min-max latency, there is a cyclic solution that is optimal.*

It is easy to see that there exists a cyclic solution that is a 2-approximation: take an optimal schedule with latency L , and at time L move the robots back to their respective starting positions at time 0, and repeat. The challenge lies in getting an approximation factor smaller than 2, which we achieve in Section 3 where we show that there is a cyclic solution that is a $2(1 - 1/k)$ approximation.

3 Turning an Optimal Solution into a Cyclic Solution

The main goal of this section is to prove the following theorem.

► **Theorem 2.** *Let L be the latency in an optimal solution to the k -robot patrol-scheduling problem in a metric space (P, d) . There is a cyclic solution with latency at most $2(1 - 1/k)L$.*

We prove the theorem by considering an optimal (potentially “chaotic”) solution and turning it into a cyclic solution. This is done by first identifying a certain set of “bottleneck” sites within a time interval of length L , then cutting the schedules into smaller pieces, and then gluing them together to obtain the final cyclic solution. This will require some graph-theoretic tools as well as several new ideas (see Appendix in the full-version paper).

Below we sketch the main ideas of the proof; the full proof can be found in Appendix of the full-version paper.

3.1 Bidirectional sweep to find “bottleneck” sites

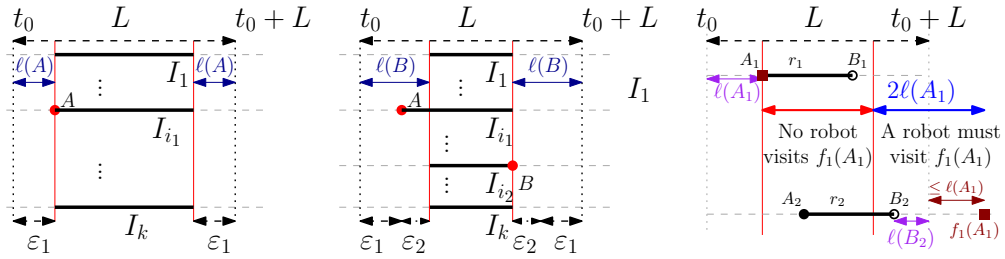
Consider an optimal patrol schedule with latency L , and consider a time interval $\mathcal{I} := [t_0, t_0 + L]$ for an arbitrary $t_0 > 2L$. By our assumptions, every site is visited at least once within this time interval. We assign a time interval $I_i \subseteq \mathcal{I}$ to every robot r_i . Initially $I_i = \mathcal{I}$. To identify the important sites that are visited by the robots, using a process that we will describe shortly, we will *shrink* each I_i . Shrinking is done by moving the left and right endpoints of I_i “inward” at the same speed. This will be done in multiple stages and at the end of each stage, an endpoint of some intervals could become *fixed*; a fixed endpoint does not move anymore during the following stages. When both endpoints of I_i are fixed, we have found the final shrunken interval for r_i . Initially, all the endpoints are *unfixed*. For an interval $I_i = [t_i, t'_i]$, shrinking I_i by some value $\varepsilon \geq 0$ yields the interval $[t_i + \varepsilon\varphi_1, t'_i - \varepsilon\varphi_2]$ where φ_1 (resp. φ_2) is 1 if the left (resp. right) endpoint of I_i is unfixed, otherwise it is 0. Note that an interval $[x, y]$ with $x > y$ is considered *empty* (i.e., an empty set) and thus shrinking an interval by a large enough value will yield an empty interval (assuming at least one endpoint is unfixed).

The invariant. We maintain the invariant that at the beginning of each stage of the shrinking process, all the sites are visited during the shrunken intervals, i.e., for every site $s \in P$, there exists a robot r_i , and a value $t \in I_i$ such that $f_i(t) = s$. Observe that the invariant holds at the beginning of the first stage of the shrinking process.

The shrinking process. Consider the j -th stage of the shrinking process. Let $\varepsilon_j \geq 0$ be the largest (supremum) number such that shrinking all the intervals by ε_j respects the invariant. If ε_j is unbounded, then this is the last stage; every interval I_i that has an unfixed endpoint is reduced to an *empty interval* and we are done with shrinking, meaning, the shrinking process has yielded some $k' \leq k$ intervals with both endpoints fixed, and $k - k'$ empty intervals. Otherwise, ε_j is bounded and well-defined as the invariant holds for $\varepsilon_j = 0$. With a slight abuse of the notation, let I_1, \dots, I_k be the intervals shrunken by ε_j . See Figure 2(left).

Since ε_j is the largest value that respects our invariant, it follows that there must be at least one interval I_{i_j} and at least one of its endpoints t_j such that at time t_j , the robot r_{i_j} visited the site $f_{i_j}(t_j)$ and this site is not visited by any other robot in the interior of their time intervals. Now this endpoint of I_{i_j} is marked as fixed and we continue to the next stage.

For a fixed endpoint A , let $\ell(A)$ be the distance of A to the corresponding boundary of the unshrunk interval. More precisely, if A is a left endpoint then the position of A on the time axis is $t_0 + \ell(A)$, and if A is a right endpoint then this position is $t_0 + L - \ell(A)$. With our notation, if A was discovered at stage j , then $\ell(A) = \varepsilon_1 + \dots + \varepsilon_j$.



■ **Figure 2** (left) A is fixed at stage 1. (middle) B is fixed at stage 2. (right) By the property of the shrinking process, the site visited at A_1 is not visited by any robot within the red time interval but since the site has latency at most L , it must be visited by some robot in the blue interval.

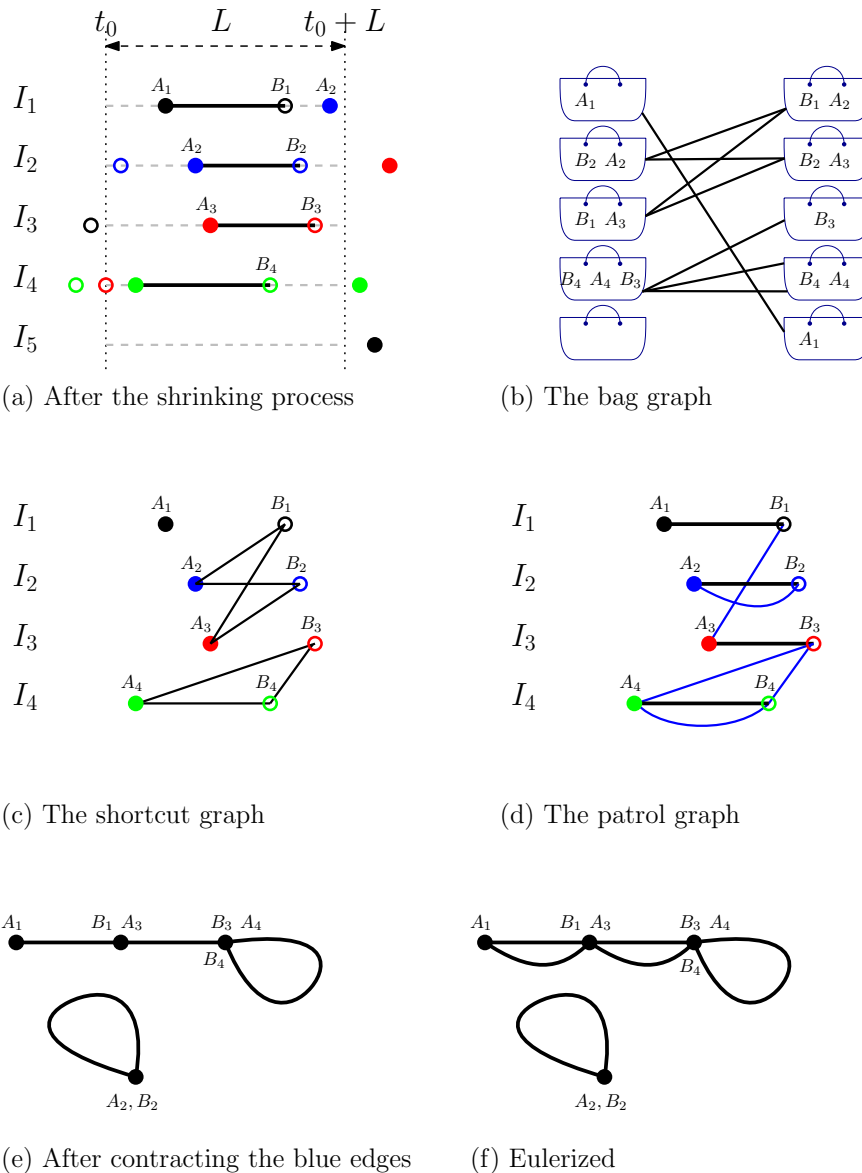
3.2 Patrol graph, shortcut graph, and bag graph

Shortcutting idea. Figure 2 (right) explains the crucial property of our shrinking process: Robot r_1 visits the site $p = f_1(A_1)$ at time A_1 (which corresponds to the left endpoint of the interval I_1) but to keep the latency of p at most L , p must be visited by another robot, say r_2 , sometime in the interval $[t_0 + L - \ell(A_1), t_0 + L + \ell(A_1)]$, shown in blue in the figure. For the moment, assume the right endpoint of the interval of r_2 is a fixed point B_2 and r_2 visits a site $p' = f_2(B_2)$ at time B_2 . This implies that the distance between p and p' is at most $\ell(A_1) + \ell(B_2)$. Now observe that we can view this as a “shortcut” between endpoints p and p' : for example, r_2 can follow its own route from A_2 to B_2 , then take the shortcut to A_1 , and then follow r_1 ’s route to B_1 . The extra cost of taking the shortcut, which is $\ell(A_1) + \ell(B_2)$, can also be charged to the two “shrunk” pieces of the two intervals (the purple intervals in the picture). Our main challenge is to show that these shortcuts can be used to create a cyclic solution with only a small increase in the latency.

To do that, we will define a number of graphs associated with the shrunken intervals. We define a *patrol graph* \mathcal{P} , a *bag graph* \mathcal{B} and a *shortcut graph* \mathcal{S} . The first two are multigraphs, whereas the shortcut graph is a simple graph.

For examples see Figure 3 on page 7 and its discussion on page 8.

We start with the bag graph and the shortcut graph. We first shrink the intervals as described previously. To define these graphs, consider $2k$ conceptual *bags*, two for each interval (including the empty intervals). More precisely, for each interval we have one *left bag* and one *right bag*. The bags are the vertices of the bag graph \mathcal{B} (Figure 3(b)). The vertices of the shortcut graph \mathcal{S} are the endpoints of the non-empty intervals. To define the edges of the two graphs, we use *placements*. We will present the details below but basically, every endpoint of a non-empty interval will be placed in two bags, one in some left bag β_1 and another time in some right bag β_2 . After this placement, we add an edge in the bag graph between β_1 and β_2 . Once all the endpoints have been placed, we add edges in the shortcut graph between every two endpoints that have been placed in the same bag (Figure 3(c)).



■ **Figure 3** Examples of bag, shortcut and patrol graph.

An example of a bag and shortcut graphs. An example is shown in Figure 3. In part (a), we have four non-empty intervals $I_1 = [A_1, B_1]$, $I_2 = [A_2, B_2]$, $I_3 = [A_3, B_3]$, $I_4 = [A_4, B_4]$ and an empty interval I_5 (we will later explain the second appearance of each endpoint in this picture and for now the reader can ignore the “floating” endpoints). An example of a bag graph is shown in Figure 3(b): Every endpoint is placed twice (once in some left bag and one in some right bag). E.g., A_1 is placed in the top-left bag and the bottom-right bag and thus the two bags are connected in the bag graph. Similarly, B_1 is placed in two bags, once at the top-right bag and the other time at the mid-left bag. In part (c) of the figure, one can see the shortcut graph in which two endpoints are connected if and only if they are placed in the same bag. Also, this is a simple graph and despite the fact that A_4 and B_4 are placed together in two different bags, they are still connected once in the shortcut graph.

Initially, all the bags are empty. For every non-empty interval $I_1 = [A_1, B_1]$, we place the left endpoint of I_1 in its own left bag and the right endpoint of I_1 in its own right bag. This is the first placement. For the second placement, consider a non-empty interval I_{i_1} and its left endpoint A . The position of A on the time interval is $t = t_0 + \ell(A)$. See Figure 2(right). By our assumptions, the robot r_{i_1} visits the site $p = f_{i_1}(t)$ at time t . Consider the stage of our shrinking process when A gets fixed. For this to happen, the site p cannot be visited by any robot in the time interval $(t_0 + \ell(A), t_0 + L - \ell(A))$ (the red interval in Figure 2(right)), as otherwise, we could either shrink all the intervals by an infinitesimal additional amount or some other endpoint would have been fixed. On the other hand, this site has latency at most L , so it must be visited by another robot in the time interval $(t, t + L] = (t_0 + \ell(A), t_0 + \ell(A) + L]$. This means that the robot r_j that visits p earliest in this interval must do so within the time interval $[t_0 + L - \ell(A), t_0 + L + \ell(A)]$ (the blue interval in Figure 2(right)). Note that r_j could be any of the robots, including r_{i_1} itself. We now place A in the right bag of I_j .

A very similar strategy is applied to the right end point of I_1 ; for details see Appendix of the full-version paper, where we also prove the following properties.

- **Lemma 3.** *The bag graph \mathcal{B} and the shortcut graph \mathcal{S} have the following properties.*
- (a) \mathcal{B} is a bipartite graph.
 - (b) \mathcal{S} is isomorphic to the line graph of \mathcal{B} .
 - (c) Let \mathcal{B}' be a connected component of \mathcal{B} . If none of the vertices of \mathcal{B}' belong to empty-intervals, then the number of vertices of \mathcal{B}' is equal to its number of edges.
 - (d) Let \mathcal{S}' be a connected component of \mathcal{S} . If \mathcal{S}' has a vertex v of degree one, then it must be the case that v corresponds to an endpoint of a non-empty interval that has been placed (alone) in a bag of an empty interval.

The patrol graph. The above lemma, combined with the graph theoretical tools that we outline in Appendix allows us to define the patrol graph \mathcal{P} . Here, we only give an outline, and for the full details see Appendix of the full version paper. An example of a patrol graph is shown in Figure 3(d). Initially, the patrol graph, \mathcal{P} , consists of k' isolated black edges, one for each non-empty interval. Observe that both \mathcal{P} and the shortcut graph \mathcal{S} have the same vertex set (endpoints of the non-empty intervals). We add a subset of the edges of the shortcut graph to \mathcal{P} . Let us consider an “easy” case to illustrate the main idea.

An easy case. Assume \mathcal{B} is connected and that it has an even number of edges. In this case, we can in fact prove that an optimal cyclic solution exists. Recall that \mathcal{S} is the line graph of \mathcal{B} and it is known that the line graph of a connected graph with even number of edges, has a perfect matching. Thus, we can find a perfect matching M as a subset of edges of \mathcal{S} . Add M to \mathcal{P} as “blue” edges. Now, every vertex of \mathcal{P} is adjacent to a blue and a black edge and thus

\mathcal{P} decomposes into a set of “bichromatic” cycles, i.e., cycles with alternating black-blue edges. With a careful accounting argument, we can show that this indeed yields a cyclic solution without increasing the latency of any of the sites. We have already mentioned the main idea under the “shortcutting idea” paragraph, at the beginning of the section. Specifically, we will use the following lemma.

► **Lemma 4.** *Consider two adjacent vertices v and w in the shortcut graph. This means that there are two non-empty intervals I_1 and I_2 such that v corresponds to an endpoint A of I_1 and w corresponds to an end point B of I_2 and A and B are placed in the same bag. Let s_1 be the site visited at A during I_1 and s_2 be the site visited at B on I_2 . Then, we have $d(s_1, s_2) \leq \ell(A) + \ell(B)$.*

Black edges represent the routes of the robots, and blue edges are the shortcuts that connect one route to another. So in this easy case, once the patrol graph has decomposed into bichromatic cycles, we turn each cycle into one closed route (i.e., cycle) using the shortcuts. All the robots that correspond to the black edges are placed evenly on this cycle. Since by our invariant all the sites are visited at some time on the black edges, it follows that the robots visit all the sites. A careful accounting argument shows that the cost of taking the shortcuts is upper bounded by the value $\ell(\cdot)$ of the end points involved. Since the values $\ell(\cdot)$ correspond to the length of the sub-paths of the robots that is missing due to our shrinking process, one can show that the latency does not increase at all.

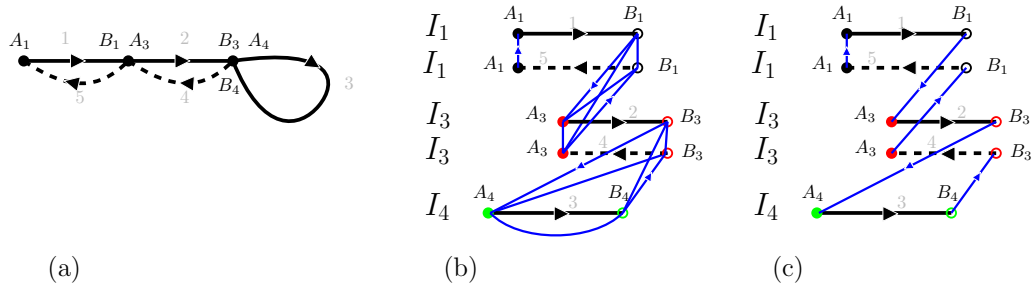
Unfortunately, \mathcal{B} can have connected components with odd number of edges. Nonetheless, in all cases we can build a particular patrol graph, \mathcal{P} , with the following properties.

► **Lemma 5.** *The patrol graph \mathcal{P} consists of k' pairwise non-adjacent black edges and a number of blue edges. Any blue edge (v, w) in \mathcal{P} corresponds to an edge in the shortcut graph \mathcal{S} . Furthermore, the set of blue edges can be decomposed into a matching and a number of triangles. In addition, any vertex of \mathcal{P} that is not adjacent to a blue edge can be charged to a bag of an empty interval.*

The idea covered in the above “easy case” works because it covers the black edges of \mathcal{P} with bichromatic edge-disjoint cycles and each cycle becomes a cyclic route. Unfortunately, in general \mathcal{P} might not have the structure that would allow us to do this. Here, we only outline the steps we need to overcome this: we consider each connected component, \mathcal{P}_i , of \mathcal{P} . We first contract blue edges of \mathcal{P}_i to obtain a contracted patrol graph, \mathcal{P}_i^c , (Figure 3(e)) then we eulerize it (Figure 3(f)), meaning, we duplicate a number of black edges such that the resulting graph is Eulerian. This yields us an Eulerized contracted patrol graph, \mathcal{P}_i^{Ec} (Figure 4(a)). Next, we put the contracted blue edges back in \mathcal{P}_i^{Ec} which gives us the final patrol graph (Figure 4(b)). In this final graph, we can show that we can cover the black edges using bichromatic edge disjoint cycles where each connected component of the final graph turns into one cycle (Figure 4(c)); this yields us a cyclic solution. However, the duplicated black edges represent routes of robots that need to be traversed twice to obtain the cyclic solution. This leads us to the final challenge: how to allocate the robots to the resulting cycles to minimize the latency. With some careful accounting and considering a few cases, we can show that this can be done in such a way that the resulting cyclic solution has latency at most $2L(1 - 1/k)$. We do this in Appendix of the full-version paper, proving Theorem 2.

4 Cyclic Solutions

In this section we show how to approximate an optimal cyclic solution to the patrol scheduling problem for k robots in a metric space (P, d) . We start with some notation and basic observations.



■ **Figure 4** (a) An Eulerized contracted patrol graph (ECPG). Duplicated edges are drawn with dashed lines. The edges are directed and numbered according to an Euler tour. (b) The same ECPG after “uncontracting” the blue edges. The duplicated routes (i.e., the routes that will be traversed twice) are shown with dashed lines. The corresponding Euler tour is marked and numbered. The shortcuts are taken in the correct direction. (c) The bichromatic cycle gives us a cyclic route. It shows how the robots can travel along it.

For a subset $Q \subseteq P$, let $\text{TSP}(Q)$ denote an optimal TSP tour of Q and let $\text{tsp}(Q)$ denote its total length. Let $\text{MST}(Q)$ denote a minimum spanning tree of Q . Now consider a partition $\Pi = \{P_1, \dots, P_t\}$ of P , where each subset P_i is assigned k_i robots such that $\sum_{i=1}^t k_i = k$. A *cyclic solution* for this partition and distribution of robots is defined as follows. For each P_i there is a cycle C_i such that the k_i robots assigned to P_i start evenly spaced along C_i and then traverse C_i at maximum speed in the same direction. Hence, the latency L of such a cyclic solution satisfies $L \geq \max_i(\text{tsp}(P_i)/k_i)$, with equality if $C_i = \text{TSP}(P_i)$ for all i .

To prove the main theorem of this section we need several helper lemmas. Let $\Pi = (P_1, \dots, P_t)$ be a partition of P and let $E \subseteq P \times P$ be a set of edges. The *coarsening* of Π with respect to E is the partition Π' of P given by the connected components of the graph $(\bigcup_i \text{MST}(P_i)) \cup E$.

► **Lemma 6.** *Let S be a cyclic solution with partition $\Pi = (P_1, \dots, P_t)$ and latency L . Let $\Pi' = (P'_1, \dots, P'_t)$ be the coarsening of Π with respect to an edge set E of total length ℓ . Then there is a cyclic solution S' with partition Π' and latency L' such that $L' \leq L + \ell$.*

Proof. Let C_1, \dots, C_t be the cycles used in S . Consider a subset $P'_i \in \Pi'$, and assume without loss of generality that P'_i is the union of the subsets P_1, \dots, P_s from Π . Then there is a set $E_i \subseteq E$ of $k - 1$ edges such that $(\bigcup_{j=1}^s C_j) \cup E_i$ is connected. Moreover, there is a cycle C'_i covering all sites in P'_i traversing the edges of each C_j once and the edges of E_i twice. Hence,

$$\|C'_i\| = \sum_{j=1}^s \|C_j\| + 2 \cdot \|E_i\|,$$

where $\|\cdot\|$ denotes the total length of a set of edges. Since the latency in S is L , we know that $\|C_j\| \leq k_j L$. Hence, using $\sum_{j=1}^s k_j \geq 2$ robots for the cycle C'_i , the latency for the sites in P'_i is at most

$$\frac{\|C'_i\|}{\sum_{j=1}^s k_j} = \frac{\sum_{j=1}^s \|C_j\| + 2 \cdot \|E_i\|}{\sum_{j=1}^s k_j} \leq \frac{\sum_{j=1}^s k_j L + 2 \cdot \|E_i\|}{\sum_{j=1}^s k_j} \leq L + \|E_i\| \leq L + \ell.$$

Thus the latency for any subset $P'_i \in \Pi'$ is at most $L + \ell$. ◀

► **Lemma 7.** *Let L^* be the latency of an optimal cyclic solution. For any $\varepsilon > 0$, there exists a cyclic solution with partition $\Pi = (P_1, \dots, P_t)$ and latency $L < (1 + \varepsilon)L^*$ such that for any pair $i \neq j$ we have $d(P_i, P_j) > \varepsilon \cdot L^*/k$, where $d(P_i, P_j) := \min\{d(x, y) : x \in P_i \text{ and } y \in P_j\}$.*

Proof. Let S^* be an optimal solution with partition $\Pi^* = (P_1^*, \dots, P_q^*)$, where $q \leq k$. Let E_{short} be the set of all edges of the complete graph of the metric space with length at most $\varepsilon L^*/k$. Let $\Pi = (P_1, \dots, P_t)$ be the partition obtained by coarsening Π^* with respect to E_{short} , and let $E^* \subseteq E_{\text{short}}$ be a minimal subset such that coarsening Π^* with E^* gives the same partition Π . Observe that as $q \leq k$, we have $|E^*| \leq k - 1$. Lemma 6 implies that there is a cyclic solution S with partition Π and latency at most

$$L^* + |E^*| \cdot (\varepsilon L^*/k) < (1 + \varepsilon)L^*.$$

Moreover, since Π is a coarsening of Π^* with respect to E_{short} , the pairwise distance between any two sets of Π is larger than $\varepsilon L^*/k$. ◀

► **Lemma 8.** *Suppose there is a cyclic solution of latency L for a given metric space (P, d) and k robots. Then $\text{MST}(P)$ has fewer than $k(1 + 1/\alpha)$ edges of length more than αL , for any $0 < \alpha \leq 1$.*

Proof. Let C_1, \dots, C_q be the cycles in the given cyclic solution of latency L , let k_i denote the number of robots assigned to C_i , and let $P_i \subset P$ be the sites in C_i . Let E be a subset of $q - 1 \leq k - 1$ edges from $\text{MST}(P)$ such that $(\bigcup_{i=1}^q C_i) \cup E$ is connected. Then

$$\sum_{i=1}^q \|C_i\| > \|\text{MST}(P)\| - \|E\| = \|\text{MST}(P) \setminus E\|$$

Since $\|C_i\| \leq k_i L$, we have $\sum_{i=1}^q \|C_i\| \leq kL$. Hence, $\|\text{MST}(P) \setminus E\| < kL$, which implies that $\text{MST}(P) \setminus E$ contains less than k/α edges of length more than αL . Including the edges in E , we thus know that $\text{MST}(P)$ has less than $k(1 + 1/\alpha)$ edges of length more than αL . ◀

► **Theorem 9.** *Suppose we have a γ -approximation algorithm for TSP in a metric space (P, d) , with running time $\tau_\gamma(n)$, and an algorithm for computing an MST that runs in time $T'(n)$. Then there is a $(1 + \varepsilon)\gamma$ -approximation algorithm for finding a minimum-latency cyclic patrol schedule with k robots that runs in $T'(n) + (O(k/\varepsilon))^k \cdot \tau_\gamma(n)$ time.*

Proof. Let L^* be the latency in an optimal cyclic solution. By Lemma 7 there is a solution S with latency $(1 + \varepsilon)L^*$ and partition $\Pi = \{P_1, \dots, P_t\}$ such that $d(P_i, P_j) > \varepsilon L^*/k$ for all $i \neq j$. Let E be the set of edges of $\text{MST}(P)$ with length more than $\varepsilon L^*/k$, and let T_1, \dots, T_z be the forest obtained from $\text{MST}(P)$ by removing E . Let $V(T_j)$ denote the sites in T_j . For any j we have $V(T_j) \subseteq P_i$ for some i . Otherwise, there would exist two sites $p, q \in V(T_j)$ that are neighbors in T_j but stay in different sets in Π . This would lead to a contradiction: the former implies $d(p, q) \leq \varepsilon L^*/k$ while the later implies $d(p, q) > \varepsilon L^*/k$. Thus Π is a coarsening of $\{V(T_1), \dots, V(T_z)\}$ with respect to some subset of E .

By Lemma 8, the number of edges of $\text{MST}(P)$ longer than $\varepsilon L^*/k$ is at most $k(1 + \frac{k}{\varepsilon})$. That is, the heaviest $k(1 + \frac{k}{\varepsilon})$ edges of $\text{MST}(P)$ are a superset of the set E from above. Thus we can find the partition Π from above by first computing $\text{MST}(P)$, removing the heaviest $k(1 + \frac{k}{\varepsilon})$ edges, and then trying all coarsenings determined by subsets of the removed edges. Given a γ -approximation for TSP, below we argue how to get a γ -approximation to the optimal cyclic solution for a given partition Π . Running this subroutine for each of the above determined partitions and taking the best solution found will thus give latency at most $(1 + \varepsilon)\gamma L^*$.

Observe that the optimal cyclic solution on a given partition $\Pi = \{P_1, \dots, P_t\}$ uses cycles determined by $\text{TSP}(P_i)$, and chooses k_i , the number of robots assigned to P_i , so as to minimize $\max_i \text{tsp}(P_i)/k_i$. Thus we can compute a γ -approximation of the optimal solution on Π by first computing a γ -approximation to $\text{TSP}(P_i)$ for all i , where $\overline{\text{tsp}}(P_i)$ denotes its corresponding value, and then selecting k'_1, \dots, k'_t so as to minimize $\max_i \overline{\text{tsp}}(P_i)/k'_i$. The latter step of determining the k'_i can be done in $O(k \log k)$ time by initially assigning one robot to each P_i , and then iteratively assigning each next robot to whichever set of the partition currently has the largest ratio. The latency of the solution we find for Π is thus

$$\max_i \left\{ \frac{\overline{\text{tsp}}(P_i)}{k'_i} \right\} \leq \max_i \left\{ \frac{\overline{\text{tsp}}(P_i)}{k_i} \right\} \leq \max_i \left\{ \frac{\gamma \cdot \text{tsp}(P_i)}{k_i} \right\} = \gamma \cdot [\text{optimal cyclic latency for } \Pi],$$

where the last inequality follows from the fact that $\overline{\text{tsp}}(P_i) \leq \gamma \cdot \text{tsp}(P_i)$ for all i .

It remains to bound the running time. For each partition Π we approximate $\text{TSP}(P_i)$ for all i , and then run an $O(k \log k)$ time algorithm to determine the robot assignment. Thus the time per partition is bounded by $\tau_\gamma(n)$, where n is the total number of sites. Here we assume that $\tau_\gamma(n) = \Omega(n \log n)$ and that $\tau_\gamma(n)$ upper bounds the time for the initial $\text{MST}(P)$ computation.

The number of partitions we consider is determined by the number of subsets of size at most k of the longest $k(1 + k/\varepsilon)$ edges of $\text{MST}(P)$, which is bounded by

$$\binom{k(1 + k/\varepsilon)}{k} \cdot 2^k,$$

as the first term bounds the number of subsets of size exactly k , and for each subset the second term accounts for the number of ways in which we can pick at most k edges from that subset. We have the following standard upper bound on binomial coefficients.

$$\binom{N}{K} \leq \left(\frac{N \cdot e}{K} \right)^K.$$

Therefore, the total number of partitions we consider is at most

$$\binom{k(1 + k/\varepsilon)}{k} \cdot 2^k \leq \left(\frac{k(1 + k/\varepsilon) \cdot e}{k} \right)^k \cdot 2^k = (2e(1 + k/\varepsilon))^k = (O(k/\varepsilon))^k.$$

Thus the total running time is $(O(k/\varepsilon))^k \cdot \tau_\gamma(n)$ as claimed. \blacktriangleleft

Recently, Karlin et al. [10] presented a $(3/2 - \delta)$ -approximation algorithm for metric TSP, where $\delta > 10^{-36}$ is a constant, thus slightly improving the classic $(3/2)$ -approximation by Christofides [5]. Furthermore TSP in \mathbb{R}^d admits a PTAS [4, 15]. Thus we have the following.

► Corollary 10. *For any fixed k , there is polynomial-time $(3/2)$ -approximation algorithm for finding a minimum-latency cyclic patrol schedule with k robots in arbitrary metric spaces, and there is a PTAS in \mathbb{R}^d for any fixed constant d .*

Theorem 2 in Section 3 and Corollary 10 together imply the following.

► Theorem 11. *For any fixed k and $\varepsilon > 0$, there is a polynomial-time $(3(1 - 1/k) + \varepsilon)$ -approximation algorithm for the k -robot patrol-scheduling problem in arbitrary metric spaces, and a polynomial-time $(2(1 - 1/k) + \varepsilon)$ -approximation algorithm in \mathbb{R}^d (for fixed d).*

5 Conclusion and Future Work

This is the first paper that presents rigorous analysis and approximation algorithms for multi-robot patrol scheduling problem in general metric spaces. There are several challenging open problems. The first and foremost is to prove or disprove the conjecture that there is always a cyclic solution that is optimal overall. Proving this conjecture will immediately provide a PTAS for the Euclidean multi-robot patrol-scheduling problem. It would also imply that the decision problem is decidable. Another direction for future research is to extend the results to the weighted setting. As has been shown for the 1-dimensional problem [1], the weighted setting is considerably harder.

References

- 1 Peyman Afshani, Mark de Berg, Kevin Buchin, Jie Gao, Maarten Löffler, Amir Nayyeri, Benjamin Raichel, Rik Sarkar, Haotian Wang, and Hao-Tsung Yang. Approximation algorithms for multi-robot patrol-scheduling with min-max latency. In *Algorithmic Foundations of Robotics XIV*, pages 107–123, 2021.
- 2 Soroush Alamdari, Elaheh Fata, and Stephen L Smith. Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations. *The International Journal of Robotics Research*, 33(1):138–154, 2014.
- 3 Esther M Arkin, Refael Hassin, and Asaf Levin. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59(1):1–18, 2006.
- 4 Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998.
- 5 Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon Univ., Pittsburgh, 1976.
- 6 George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.
- 7 Yehuda Elmaliach, Asaf Shiloni, and Gal A. Kaminka. A realistic model of frequency-based multi-robot polyline patrolling. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS '08*, pages 63–70, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.
- 8 Bruce L Golden, Subramanian Raghavan, and Edward A Wasil. *The vehicle routing problem: latest advances and new challenges*, volume 43. Springer Science & Business Media, 2008.
- 9 L. Iocchi, L. Marchetti, and D. Nardi. Multi-robot patrolling with coordinated behaviours in realistic environments. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2796–2801, September 2011. doi:10.1109/IRoS.2011.6094844.
- 10 Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric TSP. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 32–45, 2021.
- 11 M Yu Khachai and ED Neznakhina. A polynomial-time approximation scheme for the Euclidean problem on a cycle cover of a graph. *Proceedings of the Steklov Institute of Mathematics*, 289(1):111–125, 2015.
- 12 Michael Khachay and Katherine Neznakhina. Polynomial time approximation scheme for the minimum-weight k-size cycle cover problem in Euclidean space of an arbitrary fixed dimension. *IFAC-PapersOnLine*, 49(12):6–10, 2016.
- 13 M Reza Khani and Mohammad R Salavatipour. Improved approximation algorithms for the min-max tree cover and bounded tree cover problems. *Algorithmica*, 69(2):443–460, 2014.
- 14 Kin Sum Liu, Tyler Mayer, Hao-Tsung Yang, Esther Arkin, Jie Gao, Mayank Goswami, Matthew P. Johnson, Nirman Kumar, and Shan Lin. Joint sensing duty cycle scheduling for heterogeneous coverage guarantee. In *INFOCOM 2017-IEEE Conference on Computer Communications*, IEEE, pages 1–9. IEEE, 2017.

- 15 Joseph SB Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on computing*, 28(4):1298–1309, 1999.
- 16 Christos H Papadimitriou. The Euclidean travelling salesman problem is NP-complete. *Theoretical computer science*, 4(3):237–244, 1977.
- 17 D. Portugal and R. P. Rocha. On the performance and scalability of multi-robot patrolling algorithms. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 50–55, November 2011. doi:10.1109/SSRR.2011.6106761.
- 18 E. Stump and N. Michael. Multi-robot persistent surveillance planning as a vehicle routing problem. In *Automation Science and Engineering (CASE), 2011 IEEE Conference on*, pages 569–575, August 2011. doi:10.1109/CASE.2011.6042503.
- 19 Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.
- 20 Wenzheng Xu, Weifa Liang, and Xiaola Lin. Approximation algorithms for min-max cycle cover problems. *IEEE Transactions on Computers*, 64(3):600–613, 2013.
- 21 Hao-Tsung Yang, Shih-Yu Tsai, Kin Sum Liu, Shan Lin, and Jie Gao. Patrol scheduling against adversaries with varying attack durations. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1179–1188, 2019.

On Semialgebraic Range Reporting

Peyman Afshani 

Aarhus University, Denmark

Pingan Cheng 

Aarhus University, Denmark

Abstract

Semialgebraic range searching, arguably the most general version of range searching, is a fundamental problem in computational geometry. In the problem, we are to preprocess a set of points in \mathbb{R}^D such that the subset of points inside a semialgebraic region described by a constant number of polynomial inequalities of degree Δ can be found efficiently.

Relatively recently, several major advances were made on this problem. Using algebraic techniques, “near-linear space” data structures [6, 18] with almost optimal query time of $Q(n) = O(n^{1-1/D+o(1)})$ were obtained. For “fast query” data structures (i.e., when $Q(n) = n^{o(1)}$), it was conjectured that a similar improvement is possible, i.e., it is possible to achieve space $S(n) = O(n^{D+o(1)})$. The conjecture was refuted very recently by Afshani and Cheng [3]. In the plane, i.e., $D = 2$, they proved that $S(n) = \Omega(n^{\Delta+1-o(1)}/Q(n)^{(\Delta+3)\Delta/2})$ which shows $\Omega(n^{\Delta+1-o(1)})$ space is needed for $Q(n) = n^{o(1)}$. While this refutes the conjecture, it still leaves a number of unresolved issues: the lower bound only works in 2D and for fast queries, and neither the exponent of n or $Q(n)$ seem to be tight even for $D = 2$, as the best known upper bounds have $S(n) = O(n^{m+o(1)}/Q(n)^{(m-1)D/(D-1)})$ where $m = \binom{D+\Delta}{D} - 1 = \Omega(\Delta^D)$ is the maximum number of parameters to define a monic degree- Δ D -variate polynomial, for any constant dimension D and degree Δ .

In this paper, we resolve two of the issues: we prove a lower bound in D -dimensions, for constant D , and show that when the query time is $n^{o(1)} + O(k)$, the space usage is $\Omega(n^{m-o(1)})$, which almost matches the $\tilde{O}(n^m)$ upper bound and essentially closes the problem for the fast-query case, as far as the exponent of n is considered in the pointer machine model. When considering the exponent of $Q(n)$, we show that the analysis in [3] is tight for $D = 2$, by presenting matching upper bounds for uniform random point sets. This shows either the existing upper bounds can be improved or to obtain better lower bounds a new fundamentally different input set needs to be constructed.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Computational Geometry, Range Searching, Data Structures and Algorithms, Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.3

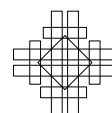
Related Version *Full Version:* <https://arxiv.org/abs/2203.07096>

Funding Supported by DFF (Det Frie Forskningsråd) of Danish Council for Independent Research under grant ID DFF-7014-00404.

1 Introduction

In the classical semialgebraic range searching problem, we are to preprocess a set of n points in \mathbb{R}^D such that the subset of points inside a semialgebraic region, described by a constant number of polynomial inequalities of degree Δ can be found efficiently. Recently, two major advances were made on this problem. First, in 2019, Agarwal et al. [5] showed for polylogarithmic query time, it is possible to build a data structure of size $\tilde{O}(n^\beta)$ space¹, where β is the number of parameters needed to specify a query polynomial. For example, for $D = 2$,

¹ $\tilde{\Omega}(\cdot), \tilde{O}(\cdot), \tilde{\Theta}(\cdot)$ notations hide $\log^{o(1)} n$ factors; $\hat{\Omega}(\cdot), \hat{O}(\cdot), \hat{\Theta}(\cdot)$ notations hide $n^{o(1)}$ factors.



a query polynomial is in the form of $\sum_{i+j \leq \Delta} a_{ij} x^i y^j \leq 0$ where a_{ij} 's are specified at the query time, and when $\Delta = 4$, β can be as large as 14 (technically, there are 15 coefficients but one coefficient can always be normalized to be 1). In this case, a major conjecture was that if this space bound could be improved to $\tilde{O}(n^D)$ (e.g., for $\Delta = 4$, from $\tilde{O}(n^{14})$ to $\tilde{O}(n^2)$). Very recently, Afshani and Cheng [3] refuted this conjecture by showing an $\tilde{\Omega}(n^{\Delta+1})$ lower bound. However, there are two major limitations of their lower bound. First, their lower bound only works in \mathbb{R}^2 , while the upper bound in [5] holds for all dimensions. Second, their lower bound only works for queries of form $y - \sum_{i=0}^{\Delta} x^i \leq 0$ and thus their lower bound does not give a satisfactory answer to the problem in the general case. For example, for $D = 2, \Delta = 4$, they show a $\tilde{\Omega}(n^5)$ lower bound whereas the current best upper bound is $\tilde{O}(n^{14})$. In general, their space lower bound is at most $\tilde{\Omega}(n^{\Delta+1})$ while the upper bound of [5] can be $\tilde{O}(n^{\Theta(\Delta^2)})$, which leaves an unsolved wide gap, even for $D = 2$. Another problem brought by [5] is the space-time tradeoff. When restricted to queries of the form $y - \sum_{i=0}^{\Delta} x^i \leq 0$, the current upper bound tradeoff is $S(n) = \tilde{O}(n^{\Delta+1}/Q(n)^{2\Delta})$ [18, 5] while the lower bound in [3] is $S(n) = \tilde{\Omega}(n^{\Delta+1}/Q(n)^{(\Delta+3)\Delta/2})$. Even for $\Delta = 2$, we observe a discrepancy between an $S(n) = \tilde{O}(n^3/Q(n)^4)$ upper and an $S(n) = \tilde{\Omega}(n^3/Q(n)^5)$ lower bound.

Here, we make progress in both lower and upper bound directions. We give a general lower bound in D dimensions that is tight for all possible values of β . Our lower bound attains the maximum possible β value $m_{D,\Delta} = \binom{D+\Delta}{D} - 1$, e.g., $\tilde{\Omega}(n^{14})$ for $D = 2, \Delta = 4$. Thus, our lower bounds almost completely settle the general case of the problem for the fast-query case, as far as the exponent of n is concerned. This improvement is quite non-trivial and requires significant new insights that are not available in [3]. For the upper bound, we present a matching space-time tradeoff for the two problems studied in [3] for uniform random point sets. This shows their lower bound analysis is tight. Since for most range searching problems, a uniform random input instance is the hardest one, our results show that current upper bound based on the classical method might not be optimal. We develop a set of new ideas for our results which we believe are important for further investigation of this problem.

1.1 Background

In range searching, the input is a set of points in \mathbb{R}^D for a fixed constant D . The goal is to build a structure such that for a query range, we can report or find the points in the range efficiently. This is a fundamental problem in computational geometry with many practical uses in e.g., databases and GIS systems. For more information, see surveys by Agarwal [14] or Matoušek [17]. We focus on a fundamental case of the problem where the ranges are semialgebraic sets of constant complexity which are defined by intersection/union/complementation of $O(1)$ polynomial inequalities of constant degree at most Δ in \mathbb{R}^D .

The study of this problem dates back to at least 35 years ago [19]. A linear space and $O(n^{1-1/D+o(1)})$ query time structure is given by Agarwal, Matoušek, and Sharir [6], due to the recent ‘‘polynomial method’’ breakthrough [15]. However, it is not entirely clear what happens to the ‘‘fast-query’’ case: if we insist on polylogarithmic query time, what is the smallest possible space usage? Early on, some believed that the number of parameters plays an important role and thus $\tilde{O}(n^\beta)$ space could be a reasonable conjecture [17], but such a data structure was not found until 2019 [5]. However, after the ‘‘polynomial method’’ revolution, and specifically after the breakthrough result of Agarwal, Matoušek and Sharir [6], it could also be reasonably conjectured that $\tilde{O}(n^D)$ could also be the right bound. However, this was refuted recently by Afshani and Cheng [3] who showed that in 2D, and for

polynomials for the form $y - \sum_{i=0}^{\Delta} x^i \leq 0$, there exists an $\tilde{\Omega}(n^{\Delta+1})$ space lower bound for data structures with query time $\tilde{O}(1)$. However, this lower bound does not go far enough, even in 2D, where a semialgebraic range can be specified by bivariate monic polynomial inequalities² of form $\sum_{i,j:i+j \leq \Delta} a_{ij} x^i y^j \leq 0$ with $a_{\Delta 0} = -1$. In this case, β can be as large as $m_{2,\Delta} = \binom{\Delta+2}{2} - 1 = \Theta(\Delta^2)$, and much larger than $\Delta + 1$ even for moderate Δ (e.g., for $\Delta = 4$, “5” versus “14”, for $\Delta = 5$, “6” versus “20” and so on). Another main weakness is that their lower bound is only in 2D, but the upper bound [5] works in arbitrary dimensions.

The correct upper bound tradeoff seems to be even more mysterious. Typically, the tradeoff is obtained by combining the linear space and the polylogarithmic query time solutions. For simplex range searching (i.e., when $\Delta = 1$), the tradeoff is $S(n) = \tilde{O}(n^D/Q(n)^D)$ [16], which is a natural looking bound and it is also known to be optimal. The tradeoff bound becomes very mysterious for semialgebraic range searching. For example, for $D = 2$ and when restricted to queries of the form $y - \sum_{i=0}^{\Delta} x^i \leq 0$, combining the existing solutions yields the bound $S(n) = \tilde{O}(n^{\Delta+1}/Q(n)^{2\Delta})$ whereas the known lower bound [3] is $S(n) = \tilde{\Omega}(n^{\Delta+1}/Q(n)^{(\Delta+3)\Delta/2})$. One possible reason for this gap is that the lower bound construction is based on a uniform random point set, while in practice, the input can be pathological. But in general the uniform random point set assumption is not too restrictive for range searching problems. Almost all known lower bounds rely on this assumption: e.g., half-space range searching [9, 7, 8], orthogonal range searching [11, 12, 2], simplex range searching [10, 13, 1].

1.2 Our Results

Our results consist of two parts. First, we study a problem that we call “the general polynomial slab range reporting”. Formally, let $P(X)$ be a monic D -variate polynomial of degree at most Δ , a general polynomial slab is defined to be the region between $P(X) = 0$ and $P(X) = w$ for some parameter w specified at the query time. Unlike [3], our construction can reach the maximum possible parameter number $m_{D,\Delta}$. For simplicity, we use m instead of $m_{D,\Delta}$ when the context is clear. We give a space-time tradeoff lower bound of $S(n) = \tilde{\Omega}(n^m/Q(n)^{\Theta((\Delta^2+D\Delta)m)})$, which is (almost) tight when $Q(n) = n^{o(1)}$.

For the second part, we present data structures that match the lower bounds studied in the work by Afshani and Cheng [3]. We show that their lower bounds for 2D polynomial slabs and 2D annuli are tight for uniform random point sets. Our bound shows that current tradeoff given by the classical method of combining extreme solutions [18, 5] might not be tight. We shed some lights on the upper bound tradeoff and develop some ideas which could be used to tackle the problem. Our results are summarized in Table 1.

1.3 Technical Contributions

Compared to the previous lower bound in [3], we need to wrestle with many complications that stem from the algebraic geometry nature of the problem. In Section 3, we cover them in greater detail, but briefly speaking, the technical heart of the results in [3] is that “two univariate polynomials $P_1(x)$ and $P_2(x)$ that have sufficiently different leading coefficients, cannot pass close to each other for too long. However, this claim is not true for even bivariate polynomials, since $P_1(x, y)$ and $P_2(x, y)$ could have infinitely many roots in common and thus we can have $P_1(x, y) - P_2(x, y) = 0$ in an unbounded region of \mathbb{R}^2 . Overcoming this requires significant innovations.

² We define that a D -variate polynomial $P(X_1, X_2, \dots, X_D)$ is monic if the coefficient of X_2^Δ is -1 .

3:4 On Semialgebraic Range Reporting

■ **Table 1** Our Results (marked by *). Our upper bounds are for uniform random point sets.

Query Types	Lower Bound	Upper Bound
General Polynomial Slabs ($m = m_{D,\Delta} = \binom{D+\Delta}{D} - 1$) When $Q(n) = \mathring{O}(1)$	$S(n) = \mathring{\Omega} \left(\frac{n^m}{Q(n)^{\Theta(m)}} \right)^*$ $S(n) = \mathring{\Omega} (n^m)^*$	$S(n) = \tilde{O} \left(\frac{n^m}{Q(n)^{\Theta(m)}} \right)$ [18, 5] $S(n) = \tilde{O} (n^m)$ [18, 5]
2D Semialgebraic Sets ($m = m_{2,\Delta} = \binom{2+\Delta}{2} - 1$)	$S(n) = \mathring{\Omega} \left(\frac{n^m}{Q(n)^{m+m^2(m-1)-1}} \right)^*$	$S(n) = \tilde{O} \left(\frac{n^m}{Q(n)^{2m-2}} \right)$ [18, 5] $S(n) = \tilde{O} \left(\frac{n^m}{Q(n)^{3m-4}} \right)^*$
2D Polynomial Slabs	$S(n) = \mathring{\Omega} \left(\frac{n^{\Delta+1}}{Q(n)^{(\Delta+3)\Delta/2}} \right)$ [3]	$S(n) = \tilde{O} \left(\frac{n^{\Delta+1}}{Q(n)^{2\Delta}} \right)$ [18, 5] $S(n) = \tilde{O} \left(\frac{n^{\Delta+1}}{Q(n)^{(\Delta+3)\Delta/2}} \right)^*$
2D Annuli	$S(n) = \mathring{\Omega} \left(\frac{n^3}{Q(n)^5} \right)$ [3]	$S(n) = \tilde{O} \left(\frac{n^3}{Q(n)^4} \right)$ [18, 5] $S(n) = \tilde{O} \left(\frac{n^3}{Q(n)^5} \right)^*$

2 Preliminaries

In this section, we introduce some tools we will use in this paper. We will mainly use the lower bound tools used in [3]. For more detailed introduction, we refer the readers to [3].

2.1 A Geometric Lower Bound Framework

We present a lower bound framework in the pointer machine model of computation. It is a streamlined version of the framework by Chazelle [11] and Chazelle and Rosenberg [13]. In essence, this is an encapsulation of the way the framework is used in [3].

In a nutshell, in the pointer machine model, the memory is represented as a directed graph where each node can store one point and it has two pointers to two other nodes. Given a query, starting from a special “root” node, the algorithm explores a subgraph that contains all the input points to report. The size of the explored subgraph is the query time.

Intuitively, for range reporting, to answer a query fast, we need to store its output points close to each other. If each query range contains many points to report and two ranges share very few points, some points must be stored multiple times, thus the total space usage must be big. We present the framework, and refer the readers to the full version of the paper for the proof.

► **Theorem 1.** *Suppose the D -dimensional geometric range reporting problems admit an $S(n)$ space and $Q(n) + O(k)$ query time data structure, where n is the input size and k is the output size. Let $\mu^D(\cdot)$ denote the D -dimensional Lebesgue measure. (We call this D -measure for short.) Assume we can find $m = n^c$ ranges $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_m$ in a D -dimensional cube \mathbf{C}^D of side length $|l|$ for some constant c such that (i) $\forall i = 1, 2, \dots, m, \mu^D(\mathcal{R}_i \cap \mathbf{C}^D) \geq 4c|l|^D Q(n)/n$; and (ii) $\mu^D(\mathcal{R}_i \cap \mathcal{R}_j) = O(|l|^D / (n2^{\sqrt{\log n}}))$ for all $i \neq j$. Then, we have $S(n) = \mathring{\Omega}(mQ(n))$.*

2.2 A Lemma for Polynomials

Given a univariate polynomial and some positive value w , the following lemma from [3] upper bounds the length of the interval within which the absolute value of the polynomial is no more than w . We will use this lemma as a building block for some of our proofs.

► **Lemma 2** (Afshani and Cheng [3]). *Given a degree- Δ univariate polynomial $P(x) = \sum_{i=0}^{\Delta} a_i x^i$ where $|a_{\Delta}| > 0$ and $\Delta > 0$. Let w be any positive value. If $|P(x)| \leq w$ for all $x \in [x_0, x_0 + t]$ for some parameter x_0 , then $t = O((w/|a_{\Delta}|)^{1/\Delta})$.*

2.3 Useful Properties about Matrices

In this section, we recall some useful properties about matrices. We first recall some properties of the determinant of matrices. One important property is that the determinant is multilinear:

► **Lemma 3.** *Let $A = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_n]$ be a $n \times n$ matrix where \mathbf{a}_i 's are vectors in \mathbb{R}^n . Suppose $\mathbf{a}_j = r \cdot \mathbf{w} + \mathbf{v}$ for some $r \in \mathbb{R}$ and $\mathbf{w}, \mathbf{v} \in \mathbb{R}^n$, then the determinant of A , denoted $\det(A)$, is*

$$\begin{aligned} \det(A) &= \det([\mathbf{a}_1 \ \cdots \ \mathbf{a}_{j-1} \ \mathbf{a}_j \ \mathbf{a}_{j+1} \ \cdots \ \mathbf{a}_n]) \\ &= r \cdot \det([\mathbf{a}_1 \ \cdots \ \mathbf{a}_{j-1} \ \mathbf{w} \ \mathbf{a}_{j+1} \ \cdots \ \mathbf{a}_n]) \\ &\quad + \det([\mathbf{a}_1 \ \cdots \ \mathbf{a}_{j-1} \ \mathbf{v} \ \mathbf{a}_{j+1} \ \cdots \ \mathbf{a}_n]). \end{aligned}$$

One of the special types of matrices we will use is the Vandermonde matrix which is a square matrix where the terms in each row form a geometric series, i.e., $V_{ij} = x_i^{j-1}$ for all indices i and j . The determinant of such a matrix is $\det(V) = \prod_{1 \leq i < j \leq n} (x_j - x_i)$.

Given an n -tuple $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$, we can define a generalized Vandermonde matrix V^* defined by λ , where $V_{ij}^* = x_i^{\lambda_{n-j+1} + j - 1}$. The determinant of V^* is known to be the product of the determinant of the induced Vandermonde matrix V_{V^*} with $V_{ij} = x_i^{j-1}$ and the Schur polynomial $s_{\lambda}(x_1, x_2, \dots, x_n) = \sum_T x_1^{t_1} \cdots x_n^{t_n}$, where the summation is over all semistandard Young tableaux [20] T of shape λ . The exponents t_1, t_2, \dots, t_n are all nonnegative numbers. The following lemma bounds the determinant of a generalized Vandermonde matrix.

► **Lemma 4.** *Let V^* be a generalized Vandermonde matrix defined by $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$. If $n, \lambda_1 = \Theta(1)$, and for all $i, x_i = \Theta(1)$, then $\det(V^*) = \Theta(\det(V_{V^*}))$, where V_{V^*} is the induced Vandermonde matrix with $V_{ij} = x_i^{j-1}$.*

3 Lower Bound for Range Reporting with General Polynomial Slabs

In this section, we prove our main lower bound for general polynomial slabs.

► **Definition 5.** *A general polynomial slab in \mathbb{R}^D is a triple (P, a, b) where $P \in \mathbb{R}[X]$ is a degree- Δ D -variate polynomial and a, b are two real numbers such that $a < b$. A general polynomial slab is defined as $\{X \in \mathbb{R}^D : a \leq P(X) \leq b\}$. Note that due to rescaling, we can assume that the polynomial is monic.*

Before presenting our results, we first describe the technical challenges of this problem. We explain why the construction used in [3] cannot be generalized in an obvious way and give some intuition behind our lower bound construction.

3.1 Technical Challenges

Our goal is a lower bound of the form $\mathring{\Omega}(n^m/Q(n)^{\Theta(m)})$. To illustrate the challenges, consider the case $D = 2$ and the unit square $U = U^2 = [0, 1] \times [0, 1]$. To use Theorem 1, we need to generate about $\mathring{\Omega}(n^m)$ polynomial slabs such that each slab should have width approximately $\mathring{\Omega}(Q(n)/n)$, and any two slabs should intersect with area approximately $O(1/n)$. Intuitively, this means two slabs cannot intersect over an interval of length $\mathring{\Omega}(1/Q(n))$.

3:6 On Semialgebraic Range Reporting

In Lemma 2, for univariate polynomials, the observation behind their construction is that when the leading coefficients of two polynomials differ by a large number, the length of the interval in which two polynomials are close to each other is small. However, when we consider general bivariate polynomials in \mathbb{R}^2 , this observation is no longer true. For example, consider $P_1(x, y) = (x + 1)(1000x^2 + y)$ and $P_2(x, y) = (x + 1)(x^2 + 1000y)$. The leading coefficients are 1000 and 1 respectively, but since P_1, P_2 have a common factor $(x + 1)$, their zero sets have a common line. Thus any slab of width $Q(n)/n$ generated for these two polynomial will have infinite intersection area, which is too large to be useful.

At first glance, it might seem that this problem can be fixed by picking the polynomials randomly, e.g., each coefficient is picked independently and uniformly from the interval $[0, 1]$, as a random polynomial in two or more variables is irreducible with probability 1. Unfortunately, this does not work either but for some very nontrivial reasons. To see this, consider picking coefficients uniformly at random from range $[0, 1]$ for bivariate polynomials $P(x, y) = \sum_{i+j \leq \Delta} a_{ij}x^i y^j$. The probability of pick a polynomial with $0 \leq a_{0j} \leq \frac{1}{n}$ for all a_{0j} is $\frac{1}{n^{\Delta+1}}$. For such polynomials, $0 \leq P(0, y) \leq \frac{\Delta+1}{n}$ for $y \in [0, 1]$. Suppose we sampled two such polynomials, then the two slabs generated using them will contain $x = 0$ for $y \in [0, 1]$, meaning, the two slabs will have too large of an area ($\Omega(Q(n)/n)$) in common, so we cannot have that. Unfortunately, if we sample more than $n^{\Delta+1}$ polynomials, this will happen with probability close to one, and there seems to be no easy fix. A deeper insight into the issue is given below.

Map a polynomial $\sum_{i+j \leq \Delta} a_{ij}x^i y^j$ to the point $(a_{00}, a_{01}, \dots, a_{\Delta 0})$ in \mathbb{R}^m . The above randomized construction corresponds to picking a random point from the unit cube \mathbf{U} in \mathbb{R}^m . Now consider the subset Γ of \mathbb{R}^m that corresponds to reducible polynomials. The issue is that Γ intersects \mathbf{U} and thus we will sample polynomials that are close to reducible polynomials, e.g., a sampled polynomial with $a_{0j} = 0 \in [0, \frac{1}{n}]$ is close to the reducible polynomial with $a_{0j} = 0$. Pick a large enough sample and two points will lie close to the same reducible polynomial and thus they will produce a “large” overlap in the construction. Our main insight is that there exists a point \mathbf{p} in \mathbf{U} that has a “fixed” (i.e., constant) distance to Γ ; thus, we can consider a neighborhood around \mathbf{p} and sample our polynomials from there. However, more technical challenges need to be overcome to even make this idea work but it turns out, we can simply pick our polynomials from a grid constructed in the small enough neighborhood of some such point \mathbf{p} in \mathbb{R}^m .

3.2 A Geometric Lemma

In this section, we show a geometric lemma which we will use to establish our lower bound. In a nutshell, given two monic D -variate polynomials P_1, P_2 and a point $p = (p_2, p_3, \dots, p_D) \in \mathbb{R}^{D-1}$ in the $(D - 1)$ -dimensional subspace perpendicular to the X_1 -axis, we define the distance between $Z(P_1)$ ³ and $Z(P_2)$ along the X_1 -axis at point p to be $|a - b|$, where $(a, p_2, \dots, p_D) \in Z(P_1)$ and $(b, p_2, \dots, p_D) \in Z(P_2)$. In general, this distance is not well-defined as there could be multiple a and b 's satisfying the definition. But we can show that for a specific set of polynomials, a, b can be made unique and thus the distance is well-defined. For P_1, P_2 with “sufficiently different” coefficients, we present a lemma which upper bounds the $(D - 1)$ -measure of the set of points p at which the distance between $Z(P_1)$ and $Z(P_2)$ is “small”. Intuitively, this can be viewed as a generalization of Lemma 2. We first prove the lemma in 2D for bivariate polynomials, and then extend the result to higher dimensions.

³ $Z(P)$ denotes the zero set of polynomial P .

First, we define the notations we will use for general D -variate polynomials.

► **Definition 6.** Let $I^D \subseteq \{(i_1, i_2, \dots, i_D) \in \mathbb{N}^D\}^4$, $D \geq 1$, be a set of D -tuples where each tuple consists of nonnegative integers. We call I^D an index set (of dimension D). Let $X^D = (X_1, X_2, \dots, X_D)$ be a D -tuple of indeterminates. When the context is clear, we use X for simplicity. Given an index set I^D , we define

$$P(X) = \sum_{i \in I^D} A_i X^i,$$

where $A_i \in \mathbb{R}$ is the coefficient of X^i and $X^i = X_1^{i_1} X_2^{i_2} \dots X_D^{i_D}$, to be a D -variate polynomial. For any $i \in I^D$, we define $\sigma(i) = \sum_{j=1}^D i_j$. Let Δ be the maximum $\sigma(i)$ with $A_i \neq 0$, and we say P is a degree- Δ polynomial. Given a D -tuple T , we use $T_{:j}$ to denote a j -tuple by taking only the first j components of T . Also, we use notation T_j to specify the j -th component of T . Conversely, given a $(D - 1)$ -tuple t and a value v , we define $t \oplus v$ to be the D -tuple formed by appending v to the end of t .

We will consider polynomials of form

$$P(X) = X_1 - X_2^\Delta + \sum_{i \in I^D} A_i X^i,$$

where $0 \leq A_{ij} = O(\epsilon) = o(1)$ for all $\sigma(i) \leq \Delta$ except that $A_i = 0$ for $i = (0, \Delta, 0, \dots, 0)$. Intuitively, these are monic polynomials packed closely in the neighborhood of $P(X) = X_1 - X_2^\Delta$. For simplicity, we call them “packed” polynomials. We will prove a property for packed polynomials that are “sufficiently distant”. More precisely,

► **Definition 7.** Given two distinct packed degree- Δ D -variate polynomials P_1, P_2 , we say P_1, P_2 are “distant” if each coefficient of $P_1 - P_2$ has absolute value at least $\xi_D = \delta \tau^{\mathcal{B}} (\eta \tau)^{(D-2)\Delta} > 0$ if not zero for parameters $\delta, \eta, \tau > 0$ and $\eta \tau = O((1/\epsilon)^{1/\mathcal{B}})$, where $\mathcal{B} = \binom{b}{2}$ and $b = \mathbf{m}_{2,\Delta}$ is the maximum number of coefficients needed to define a monic degree- Δ bivariate polynomial.

We will use the following simple geometric observation. See the full version of the paper for the proof.

► **Observation 8.** Let P be a packed D -variate polynomial and $a = (a_1, a_2, \dots, a_D) \in Z(P)$. If $a_i \in [1, 2]$ for all $i = 2, 3, \dots, D$, then there exists a unique a_1 such that $0 < a_1 = O(1)$.

With this observation, we can define the distance between the zero sets of two polynomials along the X_1 -axis at a point in $[1, 2]^{D-1}$ of the subspace perpendicular to the X_1 axis.

► **Definition 9.** Given two packed polynomials P_1, P_2 and a point $p = (p_2, p_3, \dots, p_D) \in [1, 2]^{D-1}$, we define the distance between $Z(P_1)$ and $Z(P_2)$ at p , denoted $\pi(Z(P_1), Z(P_2), p)$, to be $|a - b|$ s.t. $a, b > 0$, and $(a, p_2, p_3, \dots, p_D) \in Z(P_1)$ and $(b, p_2, p_3, \dots, p_D) \in Z(P_2)$.

Now we show a generalization of Lemma 2 to distant bivariate polynomials in 2D.

► **Lemma 10.** Let P_1, P_2 be two distinct distant bivariate polynomials. Let $I = \{y : \pi(Z(P_1), Z(P_2), y) = O(w) \wedge y \in [1, 2]\}$, where $w = \delta/\eta^{\mathcal{B}} = o(1)$. Then $|I| = O(\frac{1}{\eta \tau})$.

⁴ In this paper, $\mathbb{N} = \{0, 1, 2, \dots\}$.

3:8 On Semialgebraic Range Reporting

Proof. We prove it by contradiction. The idea is that if the claim does not hold, then we can “tweak” the coefficients of P_2 by a small amount such that the tweaked polynomial and P_1 have \mathbf{b} common roots. Next, we show this implies that the tweaked polynomial is equivalent to P_1 . Finally we reach a contradiction by noting that by assumption at least one of the coefficients of P_1 and P_2 is not close. Let $P_1(x, y) = x - y^\Delta + \sum_{i=0}^{\Delta} \sum_{j=0}^{\Delta-i} a_{ij} x^i y^j$ and $P_2(x, y) = x - y^\Delta + \sum_{i=0}^{\Delta} \sum_{j=0}^{\Delta-i} b_{ij} x^i y^j$ where by definition all a_{ij} ’s and b_{ij} ’s are $O(\epsilon)$. Suppose for the sake of contradiction that $|I| = \omega(\frac{1}{\eta\tau})$. We pick \mathbf{b} values $y_1, y_2, \dots, y_{\mathbf{b}}$ in I s.t. $|y_i - y_j| \geq |I|/\mathbf{b}$ for all $i \neq j$. Let $x_1, x_2, \dots, x_{\mathbf{b}}$ be the corresponding values s.t. $(x_k, y_k) \in Z(P_1)$ in the first quadrant, i.e., $P_1(x_k, y_k) = 0$ for $k = 1, 2, \dots, \mathbf{b}$. Note that

$$P_1(x_k, y_k) = 0 \equiv x_k - y_k^\Delta + \sum_{i=0}^{\Delta} \sum_{j=0}^{\Delta-i} a_{ij} x_k^i y_k^j = 0 \implies x_k = y_k^\Delta - O(\epsilon),$$

since $a_{ij} = O(\epsilon)$ and $x_k, y_k = O(1)$ by Observation 8. Since $\pi(Z(P_1), Z(P_2), y_k) = O(w)$ for all $y_k \in I$, let $(x_k + \Delta x_k, y_k)$ be the points on $Z(P_2)$, we have $P_2(x_k + \Delta x_k, y_k) = P_2(x_k, y_k) + \Theta(\Delta x_k) = 0$. Since $|\Delta x_k| = O(w)$, $P_2(x_k, y_k) = \gamma_k$ for some $|\gamma_k| = O(w)$. We would like to show that we can “tweak” every coefficient b_{ij} of $P_2(x, y)$ by some value \mathbf{d}_{ij} , to turn P_2 into a polynomial Q s.t. $Q(x_k, y_k) = 0, \forall k = 1, 2, \dots, \mathbf{b}$. If so, for every pair (x_k, y_k) ,

$$\begin{aligned} Q(x_k, y_k) &= x_k - y_k^\Delta + \sum_{i=0}^{\Delta} \sum_{j=0}^{\Delta-i} (b_{ij} + \mathbf{d}_{ij}) x_k^i y_k^j \\ &= P_2(x_k, y_k) + \sum_{i=0}^{\Delta} \sum_{j=0}^{\Delta-i} \mathbf{d}_{ij} x_k^i y_k^j \\ &= \gamma_k + \sum_{i=0}^{\Delta} \sum_{j=0}^{\Delta-i} \mathbf{d}_{ij} (y_k^\Delta - O(\epsilon))^i y_k^j \\ &= \gamma_k + \sum_{i=0}^{\Delta} \sum_{j=0}^{\Delta-i} \mathbf{d}_{ij} (y_k^{i\Delta} - O(\epsilon)) y_k^j, \end{aligned}$$

where the last equality follows from $\epsilon = o(1)$ and $1 \leq y_k \leq 2$. So to find \mathbf{d}_{ij} ’s and to be able to tweak $P_2(x, y)$, we need to solve the following linear system

$$\begin{bmatrix} 1 & y_1 & y_1^2 & \cdots & y_1^{\Delta-1} & y_1^\Delta - O(\epsilon) & \cdots & y_1^{\Delta^2} - O(\epsilon) \\ 1 & y_2 & y_2^2 & \cdots & y_2^{\Delta-1} & y_2^\Delta - O(\epsilon) & \cdots & y_2^{\Delta^2} - O(\epsilon) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & y_{\mathbf{b}} & y_{\mathbf{b}}^2 & \cdots & y_{\mathbf{b}}^{\Delta-1} & y_{\mathbf{b}}^\Delta - O(\epsilon) & \cdots & y_{\mathbf{b}}^{\Delta^2} - O(\epsilon) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{d}_{00} \\ \mathbf{d}_{01} \\ \vdots \\ \mathbf{d}_{\Delta 0} \end{bmatrix} = \begin{bmatrix} -\gamma_1 \\ -\gamma_2 \\ \vdots \\ -\gamma_{\mathbf{b}} \end{bmatrix},$$

where the exponents of y_k are generated by $i\Delta + j$ for $i, j \in \{0, 1, 2, \dots, \Delta\}$, $j \neq \Delta$, and $i + j \leq \Delta$. Let us call the above linear system $A \cdot \mathbf{d} = \boldsymbol{\gamma}$.

By Lemma 3, $\det(A) = \det(A^*) + \sum_{l=1}^{\Theta(1)} \det(A_l)$, where A^* is a generalized Vandermonde matrix defined by an \mathbf{b} -tuple $\lambda = (\Delta^2 - \mathbf{b}, \dots, 0)$, and each A_l is a matrix with some columns being $O(\epsilon)$. Since $\mathbf{b} = \binom{2+\Delta}{2} - 1$ is $\Theta(1)$, by Lemma 4, we can bound $\det(A^*)$ by $\Theta(\det(V_{A^*}))$, where V_{A^*} is the induced Vandermonde matrix. Since $|y_i - y_j| = \Omega(|I|)$ for $i \neq j$, $\det(V_{A^*}) = \prod_{1 \leq i < j \leq \mathbf{b}} (y_j - y_i) = \Omega(|I|^{\mathbf{B}})$. On the other hand, for every matrix A_l , there is at least one column where the magnitude of all the entries is $O(\epsilon)$. Since all other entries are bounded by $O(1)$, by the Leibniz formula for determinants, $|\det(A_l)| = O(\epsilon) = O((\frac{1}{\eta\tau})^{\mathbf{B}})$. Since $|I|^{\mathbf{B}} = \omega((\frac{1}{\eta\tau})^{\mathbf{B}})$, we can bound $|\det(A)| = \Omega(|I|^{\mathbf{B}})$ and in particular $|\det(A)| \neq 0$

and thus the above system has a solution and the polynomial Q exists. Furthermore, we can compute $\mathbf{d} = A^{-1}\boldsymbol{\gamma} = \frac{1}{\det(A)}C \cdot \boldsymbol{\gamma}$, where C is the cofactor matrix of A . Since all entries of A are bounded by $O(1)$, then the entries of C , being cofactors of A , are also bounded by $O(1)$. Since $|\boldsymbol{\gamma}_k| = O(w)$ and $|I| = \omega(\frac{1}{\eta\tau})$, for every $k = 1, 2, \dots, \mathbf{b}$, we have $|\mathbf{d}_{i,j}| = O(w/|I|^{\mathbf{B}}) = o(w(\eta\tau)^{\mathbf{B}}) = o(\delta\tau^{\mathbf{B}})$.

However, since both $Z(P_1)$ and $Z(Q)$ pass through these \mathbf{b} points, both P_1 and Q should satisfy $A \cdot \mathbf{c}_1 = 0$ and $A \cdot \mathbf{c}_2 = 0$, where $\mathbf{c}_1, \mathbf{c}_2$ are their coefficient vectors respectively. But since $\det(A) \neq 0$, $\mathbf{c}_1 = \mathbf{c}_2$, meaning, $P_1 \equiv Q$. This means for every $i, j = 0, 1, \dots, \Delta$, where $j \neq \Delta$ and $i + j \leq \Delta$, $|a_{ij} - b_{ij}| = \mathbf{d}_{i,j} = o(\delta\tau^{\mathbf{B}})$. However, by assumption, if two polynomials are not equal, then there exists at least one c_{ij} such that they differ by at least $\delta\tau^{\mathbf{B}}$, a contradiction. So $|I| = O(\frac{1}{\eta\tau})$. ◀

We now generalize Lemma 10 to higher dimensions.

▶ **Lemma 11.** *Let P_1, P_2 be two distinct distant D -variate polynomials. Let $S = \{X : \pi(Z(P_1), Z(P_2), X) = O(w) \wedge X \in [1, 2]^{D-1}\}$, where $w = \delta/\eta^{\mathbf{B}} = o(1)$. Then $\mu^{D-1}(S) = O(\frac{1}{\eta\tau})$.*

Proof. We prove the lemma by induction. The base case when $D = 2$ is Lemma 10. Now suppose the lemma holds for dimension $D - 1$, we prove it for dimension D . Observe that we can rewrite a D -variate polynomial $P(X) = X_1 - X_2^\Delta + \sum_{i \in I^D} A_i X^i$ as $P(X) = X_1 - X_2^\Delta + \sum_{j \in I_{D-1}^D} (f_j(X_D)) X_{D-1}^j$, where $f_j(X_D) = \sum_{k=0}^{\Delta-\sigma(j)} A_{j \oplus k} X_D^k$. Consider two distinct distant D -variate polynomials $P(X) = X_1 - X_2^\Delta + \sum_{i \in I^D} A_i X^i$ and $Q(X) = X_1 - X_2^\Delta + \sum_{i \in I^D} B_i X^i$. Let f_j, g_j be the corresponding coefficients for X_{D-1}^j . Note that there exists some j such that $f_j \not\equiv g_j$ because P_1, P_2 are distinct. Let $h_j(X_D) = f_j(X_D) - g_j(X_D)$ and observe that h_j is a univariate polynomial in X_D . We show that the interval length of X_D in which $|h_j(X_D)| < \xi_{D-1}$ is upper bounded by $O(\frac{1}{\eta\tau})$ for any $h_j(X_D) \not\equiv 0$. Pick any $h_j(X_D) \not\equiv 0$ and note that this means there exists at least one coefficient of $h_j(X_D)$ that is nonzero. By assumption, each coefficient of $h_j(X_D)$ has absolute value at least ξ_D if not zero. If the constant term is the only nonzero term, then the interval length of X_D in which $|h_j(X_D)| < \xi_{D-1}$ is 0, since $|h_j(X_D)| \geq \xi_D > \xi_{D-1}$ by definition. Otherwise by Lemma 2, the interval length $|r|$ for X_D in which $|h_j(X_D)| < \xi_{D-1}$ is upper bounded by

$$|r| = O\left(\left(\frac{\xi_{D-1}}{\xi_D}\right)^{1/\Delta}\right) = O\left(\left(\frac{1}{(\eta\tau)^\Delta}\right)^{1/\Delta}\right) = O\left(\frac{1}{\eta\tau}\right).$$

Since the total number of different j 's is $\Theta(1)$, the total number of $h_j(X_D)$ is then $\Theta(1)$. So the total interval length for X_D within which there is some nonzero $h_j(X_D)$ with $|h_j(X_D)| < \delta\tau_{D-1}$ is upper bounded by $\Theta(1) \cdot O(\frac{1}{\eta\tau}) = O(\frac{1}{\eta\tau})$. Since we are in a unit hypercube, we can simply upper bound $\mu^{D-1}(S)$ by $O(\frac{1}{\eta\tau}) \cdot \Theta(1) = O(\frac{1}{\eta\tau})$. Otherwise, by the inductive hypothesis, the $(D - 2)$ -measure of S in $[1, 2]^{D-2}$ is upper bounded by $O(\frac{1}{\eta\tau})$. Integrating over all X_D , $\mu^{D-1}(S)$ is bounded by $O(\frac{1}{\eta\tau})$ in this case as well. ◀

3.3 Lower Bound for General Polynomial Slabs

Now we are ready to present our lower bound construction. We will use a set S of D -variate polynomials in $\mathbb{R}[X]$ of form:

$$P(X) = X_1 - X_2^\Delta + \sum_{i \in I^D} A_i X^i,$$

3:10 On Semialgebraic Range Reporting

where X is a D -tuple of indeterminates, I^D is an index set containing all D -tuples i satisfying $\sigma(i) \leq \Delta$, and each $A_i \in \{k\xi_D : k = \lfloor \frac{\epsilon}{2\xi_D} \rfloor, \lfloor \frac{\epsilon}{2\xi_D} \rfloor + 1, \dots, \lfloor \frac{\epsilon}{\xi_D} \rfloor\}$ for some $\xi_D = \delta\tau^{\mathcal{B}}(\eta\tau)^{(D-2)\Delta}$ to be set later, except for one special coefficient: we set $A_i = 0$ for $i = (0, \Delta, 0, \dots, 0)$. Note that every pair of the polynomials in \mathcal{S} is distant. A general polynomial slab is defined to be a triple $(P, 0, w)$ where $P \in \mathcal{S}$ and w is a parameter to be set later. We need $w = o(\epsilon)$ and $\epsilon = o(1)$.

We consider a unit cube $\mathbf{U}^D = \prod_{i=1}^D [1, 2] \subseteq \mathbb{R}^D$ and use Framework 1. Recall that to use Framework 1, we need to lower bound the intersection D -measure of each slab we generated and \mathbf{U}^D , and upper bound the intersection D -measure of two slabs.

Given a slab $(P, 0, w)$ in our construction, first note that both P and $P - w$ are packed polynomials. We define the width of $(P, 0, w)$ to be the distance between $Z(P)$ and $Z(P - w)$ along the X_1 -axis. The following lemma shows that the width of each slab we generate will be $\Theta(w)$ in \mathbf{U}^D . See the full version of the paper for the proof.

► **Lemma 12.** *Let $P_1 \in \mathcal{S}$ and $P_2 = P_1 - r$ for any $0 \leq r = O(w)$. Then $\pi(Z(P_1), Z(P_2), X) = \Theta(r)$ for any $X \in [1, 2]^{D-1}$.*

The following simple lemma bounds the $(D-1)$ -measure of the projection of the intersection of the zero set of any polynomial in our construction and \mathbf{U}^D on the $(D-1)$ -dimensional subspace perpendicular to X_1 -axis. See the full version of the paper for the proof.

► **Lemma 13.** *Let $P \in \mathcal{S}$. The projection of $Z(P) \cap \mathbf{U}^D$ on the $(D-1)$ -dimensional space perpendicular to the X_1 -axis has $(D-1)$ -measure $\Theta(1)$.*

Combining Lemma 12 and Lemma 13, we easily bound the intersection D -measure of any slab in our construction and \mathbf{U}^D .

► **Corollary 14.** *Any slab in our construction intersects \mathbf{U}^D with D -measure $\Theta(w)$.*

Combining Lemma 12 and Lemma 11, we easily bound the intersection D -measure of two slabs in our construction in \mathbf{U}^D .

► **Corollary 15.** *Any two slabs in our construction intersect with D -measure $O(\frac{w}{\eta\tau})$ in \mathbf{U}^D .*

Since there are at most $\mathbf{m} = \binom{D+\Delta}{D} - 1$ parameters for a degree- Δ D -variate monic polynomial, the number of polynomial slabs we generated is then

$$\Theta\left(\left(\frac{\epsilon}{\xi_D}\right)^{\mathbf{m}}\right) = \Theta\left(\left(\frac{n}{Q(n)^{1+2\mathcal{B}+(D-2)\Delta} 2^{((D-2)\Delta+2\mathcal{B})\sqrt{\log n}}}\right)^{\mathbf{m}}\right) = O(n^{\mathbf{m}}),$$

by setting $\delta = wQ(n)^{\mathcal{B}}$, $\eta = Q(n)$, $\tau = 2\sqrt{\log n}$, $\epsilon = \frac{1}{Q(n)^{\mathcal{B}2^{\mathcal{B}}\sqrt{\log n}}}$, and $w = c_w Q(n)/n$ for a sufficiently large constant c_w . We pick c_w s.t. each slab intersects \mathbf{U}^D with D -measure, by Corollary 14, $\Omega(w) \geq 4\mathbf{m}Q(n)/n$. By Corollary 15 the D -measure of the intersection of two slabs is upper bounded by $O(\frac{w}{Q(n)2^{\sqrt{\log n}}}) = O(\frac{1}{n2^{\sqrt{\log n}}})$. By Theorem 1, we get the lower bound $S(n) = \overset{\circ}{\Omega}\left(n^{\mathbf{m}}/Q(n)^{\mathbf{m}+2\mathbf{m}\mathcal{B}+\mathbf{m}(D-2)\Delta-1}\right)$. Thus we get the following result.

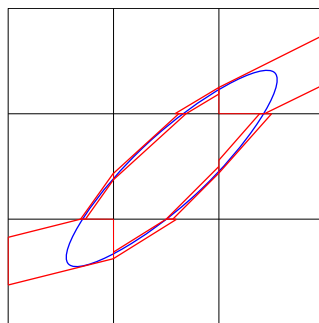
► **Theorem 16.** *Let \mathcal{P} be a set of n points in \mathbb{R}^D , where $D \geq 2$ is an integer. Let \mathcal{R} be the set of all D -dimensional generalized polynomial slabs $\{(P, 0, w) : \deg(P) = \Delta \geq 2, w > 0\}$ where $P \in \mathbb{R}[X_1, X_2, \dots, X_D]$ is a monic degree- Δ polynomial. Let \mathbf{b} (resp. \mathbf{m}) be the maximum number of parameters needed to specify a moine degree- Δ bivariate (resp. D -variate) polynomial. Then any data structure for \mathcal{P} that can answer generalized polynomial slab reporting queries from \mathcal{R} with query time $Q(n) + O(k)$, where k is the output size, must use $S(n) = \overset{\circ}{\Omega}\left(\frac{n^{\mathbf{m}}}{Q(n)^{\mathbf{m}+2\mathbf{m}\mathcal{B}+\mathbf{m}(D-2)\Delta-1}}\right)$ space, where and $\mathcal{B} = \binom{\mathbf{b}}{2}$.*

4 Data Structures for Uniform Random Point Sets

In this section, we present data structures for an input point set \mathcal{P} uniformly randomly distributed in a unit square $U = [0, 1] \times [0, 1]$ for semialgebraic range reporting queries in \mathbb{R}^2 . Our hope is that some of these ideas can be generalized to build more efficient data structures for general point sets. To this end, we show two approaches based on two different assumptions: one assumes the query curve has bounded curvature, and the other assumes bounded derivatives. We show that for any degree- Δ bivariate polynomial inequality, we can build a data structure with space-time tradeoff $S(n) = \tilde{O}(n^m/Q(n)^{3m-4})$, which is optimal for $m = 3$ [3]. When the query curve has bounded derivatives for the first Δ orders within U , this bound sharpens to $\tilde{O}(n^m/Q(n)^{((2m-\Delta)(\Delta+1)-2)/2})$, which matches the lower bound in [3] for polynomial slabs generated by inequalities of form $y - \sum_{i \leq \Delta} a_i x^i \geq 0$. Since any polynomial can be factorized into a product of $O(1)$ irreducible polynomials, and we can show that any irreducible polynomial has bounded curvature (See the full version of the paper for details), we can express the original range by a semialgebraic set consisting of $O(1)$ irreducible polynomials. We mention that both data structures can be made multilevel, then by the standard result of multilevel data structures, see e.g., [16] or [4], it suffices for us to focus on one irreducible polynomial inequality. So the curvature-based approach works for all semialgebraic sets. For both approaches, the main ideas are similar: we first partition U into a $Q(n) \times Q(n)$ grid G , and then build a set of slabs in each cell of G to cover the boundary $\partial\mathcal{R}$ of a query range \mathcal{R} . The boundaries of each slab consist of the zero sets of lower degree polynomials. We build a data structure to answer degree- Δ polynomial inequality queries inside each slab, then use the boundaries of slabs to express the remaining parts of \mathcal{R} . This lowers the degree of query polynomials, and then we can use fast-query data structures to handle the remaining parts. We assume our data structure can perform common algebraic operations in $O(1)$ time, e.g., compute roots, compute derivatives, etc.

4.1 A Curvature-based Approach

The main observation we use is that when the total absolute curvature of $\partial\mathcal{R}$ is small, the curve behaves like a line, and so we can cover it using mostly “thin” slabs, and a few “thick” slabs when the curvature is big. See Figure 1 for an example. We use the curvature as a “budget”: thin slabs have few points in them so we can afford to store them in a “fast” data structure and the overhead will be small. Doing the same with the thick slabs will blow up the space too much so instead we store them in “slower” but “smaller” data structures. The crucial observation here is that for any given query, we only need to use a few “thick” slabs so the slower query time will be absorbed in the overall query time.



■ **Figure 1** Cover an Ellipse with Slabs of Different Widths.

3:12 On Semialgebraic Range Reporting

The high-level idea is to build a two-level data structure. For the bottom-level, we build a multilevel simplex range reporting data structure [16] with query time $\tilde{O}(1) + O(k)$ and space $S(n) = \tilde{O}(n^2)$. For the upper-level, for each cell C in G and a parameter $\alpha = 2^i/Q(n)$, for $i = 0, \dots, \lfloor \log Q(n) \rfloor$, we generate a series of parallel disjoint slabs of width $\alpha/Q(n)$ such that they together cover C . Then we rotate these slabs by angle $\gamma = j/Q(n)$, for $j = 1, 2, \dots, \lfloor 2\pi Q(n) \rfloor$. For each slab we generated during this process, we collect all the points in it and build a $\tilde{O}(Q(n)\alpha) + O(k)$ query time and $\tilde{O}((n/(Q(n)\alpha))^m)$ space data structure by linearization [19] to \mathbb{R}^m and using simplex range reporting [16].

The following lemma shows we can efficiently report the points close to $\partial\mathcal{R}$ using slabs we constructed. For the proof of this lemma, we refer the readers to the full version of the paper.

► **Lemma 17.** *We can cut $\partial\mathcal{R}$ into a set \mathcal{S} of $O(Q(n))$ sub-curves such that for each sub-curve σ , we can find a set S_σ of slabs that together cover σ . Let P_σ be the subset of the input that lies inside the query and inside the slabs, i.e., $P_\sigma = \mathcal{R} \cap \mathcal{P} \cap (\cup_{s \in S_\sigma} s)$. P_σ can be reported in time $Q(n)\tilde{O}(\kappa_\sigma + 1/Q(n)) + O(|P_\sigma|)$, where κ_σ is the total absolute curvature of σ . Furthermore, for any two distinct $\sigma_1, \sigma_2 \in \mathcal{S}$, $s_1 \cap s_2 = \emptyset$ for all $s_1 \in S_{\sigma_1}, s_2 \in S_{\sigma_2}$.*

With Lemma 17, we can now bound the total query time for points close to $\partial\mathcal{R}$ by $\sum_\sigma Q(n)\tilde{O}(\kappa_\sigma + 1/Q(n)) + O(t_\sigma) = \tilde{O}(Q(n)) + O(t_1)$, where t_1 is the output size. An important observation is that after covering $\partial\mathcal{R}$, we can express the remaining regions by the boundaries of the slabs used and G , which are linear inequalities and so we can use simplex range reporting. Lemma 18 characterizes the remaining regions. See the full version of the paper for the proof.

► **Lemma 18.** *There are $O(Q(n))$ remaining regions and each region can be expressed using $O(1)$ linear inequalities. These regions can be found in time $O(Q(n))$.*

With Lemma 18, the query time for the remaining regions is $\tilde{O}(Q(n)) + O(t_2)$, where t_2 is the number of points in the remaining regions. Then the total query time is easily computed to be bounded by $\tilde{O}(Q(n)) + O(k)$, where $k = t_1 + t_2$.

To bound the space usage for the top-level data structure, note that we have $Q(n)^2$ cells, for each α , we generate $\Theta(\frac{1/Q(n)}{\alpha/Q(n)}) = \Theta(1/\alpha)$ slabs for each of the $\Theta(Q(n))$ angles. Since points are distributed uniformly at random, the expected number of points in a slab of width $\alpha/Q(n)$ in a cell C is $O(n \cdot \frac{1}{Q(n)} \cdot \frac{\alpha}{Q(n)})$. So the space usage for the top-level data structure is

$$S(n) = \sum_\alpha Q(n)^2 \cdot \Theta\left(\frac{1}{\alpha}\right) \cdot \Theta(Q(n)) \cdot \tilde{O}\left(\frac{O\left(n \cdot \frac{1}{Q(n)} \cdot \frac{\alpha}{Q(n)}\right)}{Q(n)\alpha}\right)^m = \tilde{O}\left(\frac{n^m}{Q(n)^{3m-4}}\right).$$

On the other hand, we know that the space usage for the bottom-level data structure is $\tilde{O}(n^2)$. So the total space usage is bounded by $\tilde{O}(\frac{n^m}{Q(n)^{3m-4}})$ for $m \geq 3$.

We therefore obtain the following theorem.

► **Theorem 19.** *Let \mathcal{R} be the set of semialgebraic ranges formed by degree- Δ bivariate polynomials. Suppose we have a polynomial factorization black box that can factorize polynomials into the product of irreducible polynomials in time $O(1)$, then for any $\log^{O(1)} n \leq Q(n) \leq n^\epsilon$ for some constant ϵ , and a set \mathcal{P} of n points distributed uniformly randomly in $\mathbf{U} = [0, 1] \times [0, 1]$, we can build a data structure of space $\tilde{O}(n^m/Q(n)^{3m-4})$ such that for any $\mathcal{R} \in \mathcal{R}$, we can report $\mathcal{R} \cap \mathcal{P}$ in time $\tilde{O}(Q(n)) + O(k)$ in expectation, where $m \geq 3$ is the number of parameters needed to define a degree- Δ bivariate polynomial and k is the output size.*

4.2 A Derivative-based Approach

If we assume that the derivative of $\partial\mathcal{R}$ is $O(1)$, the previous curvature-based approach can be easily adapted to get a derivative-based data structure. See the full version of the paper for details. We can even do better by using slabs whose boundaries are the zero sets of higher degree polynomials instead of linear polynomials. Using Taylor’s theorem, we show that we can cover the boundary of the query using “thin” slabs of lower degree polynomials, similar to the approach above. The full details are presented in the full version of the paper.

► **Theorem 20.** *Let \mathcal{R} be the set of semialgebraic ranges formed by degree- Δ bivariate polynomials with bounded derivatives up to the Δ -th order. For any $\log^{O(1)} n \leq Q(n) \leq n^\epsilon$ for some constant ϵ , and a set \mathcal{P} of n points distributed uniformly randomly in $\mathbf{U} = [0, 1] \times [0, 1]$, we can build a data structure which uses space $\tilde{O}(n^{\mathbf{m}}/Q(n)^{((2\mathbf{m}-\Delta)(\Delta+1)-2)/2})$ s.t. for any $\mathcal{R} \in \mathcal{R}$, we can report $\mathcal{P} \cap \mathcal{R}$ in time $\tilde{O}(Q(n)) + O(k)$ in expectation, where \mathbf{m} is the number of parameters needed to define a degree- Δ bivariate polynomial and k is the output size.*

► **Remark 21.** We remark that our data structure can also be adapted to support semialgebraic range searching queries in the semigroup model.

5 Conclusion and Open Problems

In this paper, we essentially closed the gap between the lower and upper bounds of general semialgebraic range reporting in the fast-query case at least as far as the exponent of n is concerned. We show that for general polynomial slab queries defined by D -variate polynomials of degree at most Δ in \mathbb{R}^D any data structure with query time $n^{o(1)} + O(k)$ must use at least $S(n) = \tilde{\Omega}(n^{\mathbf{m}})$ space, where $\mathbf{m} = \binom{D+\Delta}{D} - 1$ is the maximum possible parameters needed to define a query. This matches current upper bound (up to an $n^{o(1)}$ factor).

We also studied the space-time tradeoff and showed an upper bound that matches the lower bounds in [3] for uniform random point sets.

The remaining big open problem here is proving a tight bound for the exponent of $Q(n)$ in the space-time tradeoff. There is a large gap between the exponents in our lower bound versus the general upper bound. Our results show that current upper bound might not be tight. On the other hand, our lower bound seems to be suboptimal when the query time is $n^{\Omega(1)} + O(k)$. Both problems seem quite challenging, and probably require new tools.

References

- 1 Peyman Afshani. Improved pointer machine and I/O lower bounds for simplex range reporting and related problems. In *Proceedings of the Twenty-Eighth Annual Symposium on Computational Geometry*, SoCG ’12, pages 339–346, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2261250.2261301.
- 2 Peyman Afshani. A new lower bound for semigroup orthogonal range searching. In *35th International Symposium on Computational Geometry*, volume 129 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Art. No. 3, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019.
- 3 Peyman Afshani and Pigan Cheng. Lower Bounds for Semialgebraic Range Searching and Stabbing Problems. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPICs)*, pages 8:1–8:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICs.SoCG.2021.8.
- 4 Pankaj K. Agarwal. Simplex range searching and its variants: a review. In *A journey through discrete mathematics*, pages 1–30. Springer, Cham, 2017.

- 5 Pankaj K. Agarwal, Boris Aronov, Esther Ezra, and Joshua Zahl. Efficient algorithm for generalized polynomial partitioning and its applications. *SIAM J. Comput.*, 50(2):760–787, 2021. doi:10.1137/19M1268550.
- 6 Pankaj K. Agarwal, Jiří Matoušek, and Micha Sharir. On range searching with semialgebraic sets. II. *SIAM J. Comput.*, 42(6):2039–2062, 2013. doi:10.1137/120890855.
- 7 Sunil Arya, Theocharis Malamatos, and David M. Mount. On the importance of idempotence. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 564–573. ACM, New York, 2006. doi:10.1145/1132516.1132598.
- 8 Sunil Arya, David M. Mount, and Jian Xia. Tight lower bounds for halfspace range searching. *Discrete Comput. Geom.*, 47(4):711–730, 2012. doi:10.1007/s00454-012-9412-x.
- 9 Hervé Brönnimann, Bernard Chazelle, and János Pach. How hard is half-space range searching? *Discrete Comput. Geom.*, 10(2):143–155, 1993. doi:10.1007/BF02573971.
- 10 Bernard Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2(4):637–666, 1989. doi:10.2307/1990891.
- 11 Bernard Chazelle. Lower bounds for orthogonal range searching. I. The reporting case. *J. Assoc. Comput. Mach.*, 37(2):200–212, 1990. doi:10.1145/77600.77614.
- 12 Bernard Chazelle. Lower bounds for orthogonal range searching. II. The arithmetic model. *J. Assoc. Comput. Mach.*, 37(3):439–463, 1990. doi:10.1145/79147.79149.
- 13 Bernard Chazelle and Burton Rosenberg. Simplex range reporting on a pointer machine. *Comput. Geom.*, 5(5):237–247, 1996. doi:10.1016/0925-7721(95)00002-X.
- 14 Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, editors. *Handbook of discrete and computational geometry*. Discrete Mathematics and its Applications (Boca Raton). CRC Press, Boca Raton, FL, 2018. Third edition of [MR1730156].
- 15 Larry Guth and Nets Hawk Katz. On the Erdős distinct distances problem in the plane. *Ann. of Math. (2)*, 181(1):155–190, 2015. doi:10.4007/annals.2015.181.1.2.
- 16 Jiří Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10(2):157–182, 1993. doi:10.1007/BF02573972.
- 17 Jiří Matoušek. Geometric range searching. *ACM Comput. Surv.*, 26(4):421–461, 1994. doi:10.1145/197405.197408.
- 18 Jiří Matoušek and Zuzana Patáková. Multilevel polynomial partitions and simplified range searching. *Discrete Comput. Geom.*, 54(1):22–41, 2015. doi:10.1007/s00454-015-9701-2.
- 19 Andrew Chi-Chih Yao and F. Frances Yao. A general approach to d-dimensional geometric queries (extended abstract). In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 163–168. ACM, 1985. doi:10.1145/22145.22163.
- 20 A. Young. On Quantitative Substitutional Analysis. *Proc. Lond. Math. Soc.*, 33:97–146, 1901. doi:10.1112/plms/s1-33.1.97.

Intersection Queries for Flat Semi-Algebraic Objects in Three Dimensions and Related Problems

Pankaj K. Agarwal  



Department of Computer Science, Duke University, Durham, NC, USA

Boris Aronov  

Department of Computer Science and Engineering, Tandon School of Engineering,
New York University, Brooklyn, NY, USA

Esther Ezra  

School of Computer Science, Bar Ilan University, Ramat Gan, Israel

Matthew J. Katz  

Department of Computer Science, Ben Gurion University, Beer Sheva, Israel

Micha Sharir  

School of Computer Science, Tel Aviv University, Tel Aviv, Israel

Abstract

Let \mathcal{T} be a set of n planar semi-algebraic regions in \mathbb{R}^3 of constant complexity (e.g., triangles, disks), which we call *plates*. We wish to preprocess \mathcal{T} into a data structure so that for a query object γ , which is also a plate, we can quickly answer various *intersection queries*, such as detecting whether γ intersects any plate of \mathcal{T} , reporting all the plates intersected by γ , or counting them. We focus on two simpler cases of this general setting: (i) the input objects are plates and the query objects are constant-degree algebraic arcs in \mathbb{R}^3 (*arcs*, for short), or (ii) the input objects are arcs and the query objects are plates in \mathbb{R}^3 . These interesting special cases form the building blocks for the general case.

By combining the polynomial-partitioning technique with additional tools from real algebraic geometry, we obtain a variety of results with different storage and query-time bounds, depending on the complexity of the input and query objects. For example, if \mathcal{T} is a set of plates and the query objects are arcs, we obtain a data structure that uses $O^*(n^{4/3})$ storage (where the $O^*(\cdot)$ notation hides subpolynomial factors) and answers an intersection query in $O^*(n^{2/3})$ time. Alternatively, by increasing the storage to $O^*(n^{3/2})$, the query time can be decreased to $O^*(n^\rho)$, where $\rho = (2t - 3)/3(t - 1) < 2/3$ and $t \geq 3$ is the number of parameters needed to represent the query arcs.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Intersection searching, Semi-algebraic range searching, Point-enclosure queries, Ray-shooting queries, Polynomial partitions, Cylindrical algebraic decomposition, Multi-level partition trees, Collision detection

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.4

Related Version *Full Version*: <https://arxiv.org/abs/2203.10241> [3]

Funding *Pankaj K. Agarwal*: Work partially supported by NSF grants IIS-18-14493 and CCF-20-07556.

Boris Aronov: Work partially supported by NSF Grants CCF-15-40656 and CCF-20-08551, and by Grant 2014/170 from the US-Israel Binational Science Foundation.

Esther Ezra: Work partially supported by NSF CAREER under Grant CCF:AF-1553354 and by Grant 824/17 from the Israel Science Foundation.

Matthew J. Katz: Work partially supported by Grant 1884/16 from the Israel Science Foundation, and by Grant 2019715/CCF-20-08551 from the US-Israel Binational Science Foundation/US National Science Foundation.

Micha Sharir: Work partially supported by Grant 260/18 from the Israel Science Foundation.



© Pankaj K. Agarwal, Boris Aronov, Esther Ezra, Matthew J. Katz, and Micha Sharir;
licensed under Creative Commons License CC-BY 4.0

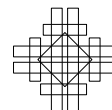
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 4; pp. 4:1–4:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Acknowledgements We thank Peyman Afshani for sharing with us problems that have motivated our study of segment-intersection searching amid spherical caps, Ovidiu Daescu for suggesting the collision-detection application that motivated some aspects of our work, and the reviewers of this work for their helpful comments.

1 Introduction

This paper studies intersection-searching problems in \mathbb{R}^3 , where both input and query objects are planar semi-algebraic regions of constant complexity (e.g., triangles, disks), which we refer to as *plates*.¹ We also consider two simpler cases of this setup: (i) the input objects are plates and the query objects are constant-degree algebraic arcs in \mathbb{R}^3 , referred to simply as *arcs*, and (ii) the input objects are arcs and the query objects are plates in \mathbb{R}^3 . Besides being interesting in their own right, the data structures for these two simpler cases form the building blocks for handling the general case. In each case, we wish to preprocess a set \mathcal{T} of input objects (plates or arcs) in \mathbb{R}^3 into a data structure that supports various *intersection queries* for a query object (again a plate or an arc) γ , where we want to determine whether γ intersects any object of \mathcal{T} (*intersection-detection* queries), report all objects of \mathcal{T} that γ intersects (*intersection-reporting* queries), count the number of objects of \mathcal{T} that γ intersects (*intersection-counting* queries), or, when the query object is a directed arc γ , report the first input object intersected by γ (*ray-shooting* queries). Intersection queries arise in many applications, including robotics, computer-aided design, and solid modeling.

Notwithstanding a considerable amount of work on segment-intersection or ray-shooting queries amid triangles in \mathbb{R}^3 (see, e.g., the survey by Pellegrini [23]), little is known about more general intersection queries in \mathbb{R}^3 , e.g., how quickly one can answer arc-intersection queries amid triangles in \mathbb{R}^3 , or triangle-intersection queries amid arcs in \mathbb{R}^3 . The present work makes significant and fairly comprehensive progress on the design of efficient solutions to general intersection-searching problems in \mathbb{R}^3 .

1.1 Related work

The general intersection-searching problem asks to preprocess a set \mathcal{O} of geometric objects in \mathbb{R}^d , so that one can quickly report or count all objects of \mathcal{O} intersected by a query object γ , or just test whether γ intersects any object of \mathcal{O} at all. One may also want to perform some other aggregate operations on these objects (see [2] for a general framework). Intersection searching is a generalization of range searching (in which the input objects are points) and point enclosure queries (in which the query objects are points).

A popular approach to answering intersection queries is to write a first-order formula for the intersection condition between an input object and a query object. Using quantifier elimination, intersection queries can be reduced to *semi-algebraic range* queries, by working in *object space*, where each input object $O \in \mathcal{O}$ is mapped to a point \hat{O} and a query object γ is mapped to a semi-algebraic region $\hat{\gamma}$, such that $\hat{\gamma}$ contains a point \hat{O} if and only if γ intersects the corresponding input object O . Alternatively, the problem can be reduced to a *point-enclosure* query, by working in *query space*, where now each input object O is mapped to a semi-algebraic region \tilde{O} and each query object γ is mapped to a point $\tilde{\gamma}$, so that $\tilde{\gamma}$ lies in \tilde{O} if and only if γ intersects O . The first approach leads to a linear-size data structure

¹ Roughly speaking, a semi-algebraic set in \mathbb{R}^d is the set of points in \mathbb{R}^d satisfying a Boolean predicate over a set of polynomial inequalities; the complexity of the predicate and of the set is defined in terms of the number of polynomials involved and their maximum degree. See [11] for details.

with sublinear query time, and the second approach leads to a large-size data structure with logarithmic or polylogarithmic query time; see, e.g., [6, 8, 14, 20, 26] for the first approach and [4, 12] for the second one.

The performance of these data structures depends on the number of parameters needed to specify the input and query objects. We refer to these numbers as the *parametric dimension* (or the number of *degrees of freedom* (dof)) of the input and query objects, respectively. Sometimes the performance can be improved using a *multi-level data structure*, where each level uses a lower-dimensional sub-predicate [2]. One can also combine these two approaches to obtain a query-time/storage trade-off. For example, using standard techniques (such as in [22]), a ray-shooting or segment-intersection query amid n triangles in \mathbb{R}^3 can be answered in $O^*(n^{3/4})$ time using $O^*(n)$ storage, in $O(\log n)$ time using $O^*(n^4)$ storage, or in $O^*(n/s^{1/4})$ time using $O^*(s)$ storage,² for $n \leq s \leq n^4$, by combining the first two solutions [22, 23]. As in the abstract, the $O^*(\cdot)$ notation hides subpolynomial factors, e.g., of the form $O(n^\varepsilon)$, for arbitrarily small $\varepsilon > 0$, and their coefficients which depend on ε . A similar multi-level approach yields data structures in which a ray-shooting query among n planes or spheres in \mathbb{R}^3 can be answered in $O^*(n/s^{1/3})$ time using $O^*(s)$ storage, for $n \leq s \leq n^3$ [21, 22, 23, 25].

A departure from this approach is the *pedestrian* approach for answering ray-shooting queries. For instance, given a simple polygon P with n edges, a Steiner triangulation of P is constructed so that a line segment lying inside P intersects only $O(\log n)$ triangles. A query is answered by traversing the query ray through this sequence of triangles [19]. The pedestrian approach has also been applied to polygons with holes in \mathbb{R}^2 [5, 19], to a convex polyhedron in \mathbb{R}^3 [15], and to polyhedral subdivisions in \mathbb{R}^3 [5, 10]. Some of the ray-shooting data structures combine the pedestrian approach with the above range-searching tools [1, 9, 14].

Recently, Ezra and Sharir [16] proposed a new approach for answering ray-shooting queries amid triangles in \mathbb{R}^3 , using the pedestrian approach in the context of the polynomial-partitioning scheme of Guth [17]. Roughly speaking, they construct a partitioning polynomial F of degree $O(D)$, for a sufficiently large constant D , using the algorithm in [4]. The zero set $Z(F)$ of F partitions \mathbb{R}^3 into *cells*, which are the connected components of $\mathbb{R}^3 \setminus Z(F)$. The partitioning scheme guarantees that, with a suitable choice of the degree, each cell τ is intersected by at most n/D input triangles, but for only at most n/D^2 of them their (relative) boundary intersects τ .³ These latter triangles are called *narrow* at τ , and the other intersecting triangles are called *wide*. For each cell τ , the algorithm of [16] recursively preprocesses the narrow triangles of τ and constructs a secondary data structure for the wide triangles at τ . A major technical result of [16] is to reduce a ray-shooting or intersection-detection query among wide triangles to a similar query amid a set of planes in \mathbb{R}^3 (those supporting the input triangles), and to use the fact that such a query amid planes can be answered in $O^*(n/s^{1/3})$ time when $O^*(s)$ storage is available, for any $n \leq s \leq n^3$; see [22, 23]. This leads to a data structure with $O^*(n^{3/2})$ storage and $O^*(n^{1/2})$ query time, which improves upon the earlier solution [22]. The approach of [16] can also support reporting queries in $O^*(n^{1/2} + k)$ time, where k is the output size, but, for certain technical reasons, it does not support counting queries.

² We sometimes refer to s as the “storage parameter,” to distinguish it from the actual storage being used, which is $O^*(s)$.

³ One actually has to construct two polynomials, one for ensuring the first property and one for the second property, and take their product, still a polynomial of degree $O(D)$, as the desired polynomial.

■ **Table 1** Summary of results. Storage and query time are $O^*(n^\alpha)$ and $O^*(n^\beta)$, respectively, and we specify the values of α and β for each result. The data structures for type (i) and (ii) intersection queries count the number of *intersection points* between the input objects and the query object, and not the number of input objects intersected by the query object.

* Counts the number of triangles intersected by a query triangle in $O^*(n^{5/9})$ time.

** This data structure does not extend to counting queries. In addition, the first term $\frac{2t_Q-7}{3(t_Q-3)}$ in the bound applies when t_Q is the maximum parametric dimension of the bounding *arcs* of the query plates; if each plate is bounded by a single endpoint-free curve, the first term in the bound becomes $\frac{2t_Q-3}{3(t_Q-1)}$.

Input	Query	Storage	Query Time
Plates	Arc/Curve	4/3	2/3
Plates	Arc/Curve ($t \geq 3$ dof)	3/2	$(2t-3)/3(t-1)$
Plates	Planar arc ($t \geq 4$ dof)	3/2	$(2t-7)/3(t-3)$
Plates	Circular arc	3/2	3/5
Triangles	Arc/Curve	1	4/5
Triangles	Arc/Curve	11/9	2/3
Spherical caps	Segment	5/4	3/4
Spherical caps	Segment	3/2	27/40
Segments	Plate	3/2	1/2
Arcs/Curves (t dof)	Plate	3/2	$3(t-1)/4t$
Triangles*	Triangle	3/2	1/2
Plates (t_O dof)**	Plate (t_Q dof)	3/2	$\max\{\frac{2t_Q-7}{3(t_Q-3)}, \frac{3(t_O-1)}{4t_O}\}$
Tetrahedra*	Tetrahedron	3/2	1/2

1.2 Our results

We refer to a connected path π as an (*algebraic*) *arc* if it is the restriction of a real algebraic curve $\gamma: I \rightarrow \mathbb{R}^3$ to a subinterval $[a, b] \subseteq I$. The *parametric dimension* t of π , also referred to as *the number of degrees of freedom* (dof) of π , is the number of real parameters needed to describe π . Two of these parameters specify the endpoints a and b . We assume that the degree of the curve is also bounded by t .

We present efficient data structures for three broad classes of intersection searching in \mathbb{R}^3 : (i) the input objects are plates and the query objects are arcs in \mathbb{R}^3 , (ii) the input objects are arcs and the query objects are plates in \mathbb{R}^3 , and (iii) both input and query objects are plates in \mathbb{R}^3 . Our algorithms combine the polynomial-partitioning technique of Guth [17] and of Guth and Katz [18] with some additional tools from real algebraic geometry.

For simplicity, we mostly focus on answering *intersection-detection* queries. Our data structures extend to answering *intersection-reporting* queries by spending additional $O(k)$ time, where k is the output size. For type (i) intersection queries, using the parametric-search framework of Agarwal and Matoušek [7], our data structures can also answer *arc-shooting* queries, where the goal is to find the first plate of \mathcal{T} hit by a (directed) query arc, if such a plate exists. Most of the data structures can be extended to answering *intersection-counting* queries as well – for type (i) and (ii) intersection queries, our data structures count the number of intersection points between the query arc/plate and the input plates/arcs, and for type (iii) queries, our approach can count the number of intersecting pairs if both input and query objects are triangles. Table 1 summarizes the main results of the paper. When we say that an intersection query can be answered in $O^*(t(n))$ time, we mean that detection, counting, and shooting queries can be answered in $O^*(t(n))$ time and reporting queries in $O^*(t(n) + k)$ time, where k is the output size.

Intersection-searching with arcs amid plates. We present several data structures for answering arc-intersection queries amid a set \mathcal{T} of n plates in \mathbb{R}^3 (cf. Section 2 and the full version [3]). Our first main result is an $O^*(n^{4/3})$ -size data structure that can be constructed in $O^*(n^{4/3})$ expected time and that supports arc-intersection queries in $O^*(n^{2/3})$ time. The asymptotic query time bound depends neither on the parametric dimension of the query arc nor on that of the input plates, though the coefficients hiding in the O^* -notation do depend on them. Although our high-level approach is similar to that of Ezra and Sharir [16], handling wide plates in our setup is significantly more challenging because the query object is an arc instead of a line segment. We handle wide input plates using a completely different approach that not only generalizes to algebraic arcs but also simplifies, in certain aspects, the technique of [16] for segment-intersection searching. Handling this much more general setup, using a battery of tools from range searching and real algebraic geometry, is one of the main technical contributions of this work. The most interesting among these tools is the construction of a carefully tailored *cylindrical algebraic decomposition* (CAD) (see [11, 13, 24] for details concerning this technique, which are also reviewed later in Section 3.1 in this work) of a suitable parametric space, where the CAD is induced by the partitioning polynomial.

Next, we present a data structure for answering arc-intersection queries amid wide plates within a cell of the polynomial partition. It reduces the query time by increasing the storage used. Roughly, the improvement is a consequence of using a combined primal-dual range-searching approach, where the primal part works in object space, as in the aforementioned main algorithm. The dual part works in query space, regarding the query arc γ as a t -dimensional point $\tilde{\gamma}$, where t is the parametric dimension of the query arcs. Each input plate Δ is mapped to a semi-algebraic range $\tilde{\Delta}$ in query space, and the query reduces to a point-enclosure query that determines whether $\tilde{\gamma}$ lies in any of these semi-algebraic ranges $\tilde{\Delta}$. Specifically, we build a data structure of size $O^*(n^{3/2})$ with $O^*\left(n^{\frac{2t-3}{3(t-1)}}\right)$ query time, for parametric dimensions $t \geq 3$.

Another significant contribution of this work is a general technique for reducing the parametric dimension t by 2, for *planar* query arcs, eliminating the dependence of the asymptotic query time bound on the endpoints of the arc. For example, if the query objects are circular arcs, their parametric dimension is eight (three for specifying the supporting plane, three for specifying the containing circle in that plane, and two for the endpoints). We show how to improve the query time from $O^*(n^{13/21})$ (the query time bound for $t = 8$) to $O^*(n^{3/5})$ (the bound for $t = 6$), with the same asymptotic storage complexity $O^*(n^{3/2})$. We note that $t = 6$ when the query objects are line segments in \mathbb{R}^3 ; by reducing this to $t = 4$, we get the query time $O^*(n^{5/9})$ for this case, which is slightly worse than $O^*(n^{1/2})$ in [16]. This deterioration in the performance is the cost we pay for proposing a general approach that extends to query objects being arcs, as well as to answering counting queries.

Next, if \mathcal{T} is a set of *triangles* in \mathbb{R}^3 , we present an alternative near-linear-size data structure that can answer an arc-intersection query, for a constant-degree algebraic arc, in $O^*(n^{4/5})$ time. Using this result in our main algorithm, we improve the storage size to $O^*(n^{11/9})$, keeping the query time $O^*(n^{2/3})$.

Intersection searching with plates amid arcs. Next, we present data structures for the complementary setup where the input objects are arcs and we query with a plate. We first show that we can preprocess a set \mathcal{T} of n line segments, in expected time $O^*(n^{3/2})$, into a data structure of size $O^*(n^{3/2})$, so that an intersection query with a plate can be answered in $O^*(n^{1/2})$ time. Next, we extend this result to the case where the input is a set

of n arcs of (constant degree and) parametric dimension t , and the query object remains a plate. We obtain a data structure of size $O^*(n^{3/2})$ that can answer an intersection query in $O^*\left(n^{\frac{3(t-1)}{4t}}\right)$ time; see the full version [3].

Intersection searching with plates amid plates. The above results can be used to provide simple solutions for the case where both input and query objects are plates. For simplicity, first assume that both input and query objects are triangles in \mathbb{R}^3 . We observe that if a query triangle Δ intersects an input triangle Δ' then $\Delta \cap \Delta'$ is a line segment, and each of its endpoints is either an intersection of an edge of Δ with Δ' or of Δ with an edge of Δ' . The former (resp., latter) kind of intersection can be detected using type (i) intersection queries (resp., type (ii) queries). Using $O^*(n^{3/2})$ storage, this results in the query time bound $O^*(n^{1/2})$, if we use the data structure from [16] for type (i) queries. For counting queries, we have to use our arc-intersection data structure, leading to a query time of $O^*(n^{5/9})$.

The technique can be extended to the case where both input and query objects are arbitrary plates. In this case, the boundary of a plate consists of $O(1)$ algebraic arcs of constant complexity. Let t_O and t_Q be the parametric dimensions of the boundary arcs of input and query plates, respectively. We obtain a data structure of $O^*(n^{3/2})$ size with query time $O^*(n^\rho)$, where $\rho = \max\left\{\frac{2t_Q-7}{3(t_Q-3)}, \frac{3(t_O-1)}{4t_O}\right\}$.⁴

Our data structure for the plate-plate case also works if the input and query objects are constant-complexity, not necessarily convex three-dimensional polyhedra. This is because an intersection between two polyhedra occurs when their boundaries meet, unless one of them is fully contained in the other, and the latter situation can be easily detected. We can therefore just triangulate the boundaries of both input and query polyhedra and apply the triangle-triangle intersection-detection machinery.

The case of spherical caps. Finally, we present an application of our technique to an instance where the input objects are not flat. Specifically, we show how to answer segment-intersection queries amid spherical caps (each being the intersection of a sphere with a halfspace), using either a data structure with $O^*(n^{5/4})$ storage and $O^*(n^{3/4})$ query time, or a structure with $O^*(n^{3/2})$ storage and $O^*(n^{27/40})$ query time.

2 Intersection searching with query arcs amid plates

Let \mathcal{T} be a set of n plates in \mathbb{R}^3 , and let Γ be a family of algebraic arcs that has parametric dimension t for some constant $t \geq 3$. We present algorithms for preprocessing \mathcal{T} into a data structure that can answer an arc-intersection query for an arc $\gamma \in \Gamma$ efficiently. We begin by describing a basic data structure, and then show how its performance can be improved.

2.1 The overall data structure

Our primary data structure consists of a partition tree Ψ on \mathcal{T} , which is constructed using the polynomial-partitioning technique of Guth [17]. More precisely, let $\mathcal{X} \subseteq \mathcal{T}$ be a subset of m plates and let $D > 1$ be a parameter. Using the result by Guth, a real polynomial F of

⁴ The first term $\frac{2t_Q-7}{3(t_Q-3)}$ in the bound applies only when the query plate is bounded by more than one arc, of maximum parametric dimension t_Q . When the query plates are bounded by a single endpoint-free curve (such as circular or elliptical disks) with parametric dimension t_Q , the term becomes $\frac{2t_Q-3}{3(t_Q-1)}$.

degree at most $c_1 D$ can be constructed, where $c_1 > 0$ is an absolute constant, such that each open connected component (called a *cell*) of $\mathbb{R}^3 \setminus Z(F)$ is crossed by boundary arcs (which we refer to as *edges* from now on) of at most m/D^2 plates of \mathcal{X} and by at most m/D plates of \mathcal{X} ; the number of cells is at most $c_2 D^3$ for some absolute constant $c_2 > 0$. Agarwal et al. [4] showed that such a partitioning polynomial can be constructed in $O(m)$ expected time if D is a constant. Using such polynomial partitionings, Ψ can be constructed recursively in a top-down manner as follows.

Each node $v \in \Psi$ is associated with a cell τ_v of some partitioning polynomial and a subset $\mathcal{T}_v \subseteq \mathcal{T}$. If v is the root of Ψ , then $\tau_v = \mathbb{R}^3$ and $\mathcal{T}_v = \mathcal{T}$. Set $n_v = |\mathcal{T}_v|$. We set a threshold parameter $n_0 \leq n$, which may depend on n , and we fix a sufficiently large constant D . For the basic data structure described here, we set $n_0 = n^{1/3}$; the value of n_0 will change when we later modify the structure.

Suppose we are at a node v . If $n_v \leq n_0$ then v is a leaf and we store \mathcal{T}_v at v . Otherwise, we construct a partitioning polynomial F_v of degree at most $c_1 D$, as described above, and store F_v at v . We construct a secondary data structure Σ_v^0 on \mathcal{T}_v for answering arc-intersection queries with an arc $\gamma \in \Gamma$ that is contained in $Z(F_v)$. Σ_v^0 is constructed in an analogous manner as Ψ by using the polynomial-partitioning scheme of Agarwal et al. [4], which, given a (constant) parameter $D_1 \gg D$, constructs a polynomial G of degree at most $c_1 D_1$ so that each cell of $Z(F) \setminus Z(G)$ intersects at most n_v/D_1 plates of \mathcal{T}_v and the boundaries of at most n_v/D_1^2 plates. Further details of Σ_v^0 (see [3]) are omitted from this version, and we conclude:

► **Proposition 2.1.** *For a partitioning polynomial F of sufficiently large constant degree and a set \mathcal{T} of n plates, one can construct, in $O^*(n)$ expected time, a data structure of size $O^*(n)$ that can answer an arc-intersection query with an arc contained in $Z(F)$ in $O^*(n^{2/3})$ time.*

Next, we compute (semi-algebraic representations of) all cells of $\mathbb{R}^3 \setminus Z(F_v)$ [11]. Let τ be such a cell. We create a child w_τ of v associated with τ . We classify each plate $\Delta \in \mathcal{T}_v$ that crosses τ as *narrow* (resp., *wide*) at τ if an edge of Δ crosses τ (resp., Δ crosses τ , but none of its edges does). Let \mathcal{W}_τ (resp., \mathcal{T}_τ) denote the set of the wide (resp., narrow) plates at τ . We construct a secondary data structure Σ_τ^1 on \mathcal{W}_τ , as described in Section 3 below, for answering arc-intersection queries amid the plates of \mathcal{W}_τ with arcs of Γ that lie inside τ . Σ_τ^1 is stored at the child w_τ of v . The construction of Σ_τ^1 for handling the wide plates is the main technical step in our algorithm. By Proposition 3.2 in Section 3, Σ_τ^1 uses $O^*(|\mathcal{W}_\tau|)$ space, can be constructed in $O^*(|\mathcal{W}_\tau|)$ expected time, and answers an arc-intersection query in $O^*(|\mathcal{W}_\tau|^{2/3})$ time. Finally, we set $\mathcal{T}_{w_\tau} = \mathcal{T}_\tau$, and recursively construct a partition tree for \mathcal{T}_{w_τ} and attach it as the subtree rooted at w_τ . Note that two secondary structures are attached at each node v , namely, Σ_v^1 and Σ_v^0 , for handling wide plates and for handling query arcs that are contained in $Z(F_v)$, respectively.

Denote by $S(m)$ the maximum storage used by the data structure for a subproblem involving at most m plates. For $m \leq n_0$, $S(m) = O(m)$. For $m > n_0$, Propositions 2.1 and 3.2 imply that $S(m)$ obeys the recurrence:

$$S(m) \leq c_2 D^3 S(m/D^2) + O^*(m), \tag{1}$$

where c_2 is the constant as defined above. Since the recursion terminates at $m \leq n_0 = n^{1/3}$, it can be shown that the solution to the above recurrence is $S(m) = O^*(m^{3/2}/n^{1/6} + m)$. Hence, the overall size of the data structure (for $m = n$) is $O^*(n^{4/3})$. A similar analysis shows that the expected preprocessing time is also $O^*(n^{4/3})$.

2.2 The query procedure

Let $\gamma \in \Gamma$ be a query arc. We answer an arc-intersection query, say, intersection-detection, for γ by searching through Ψ in a top-down manner. Suppose we are at a node v of Ψ . Our goal is to determine whether $\gamma_v := \gamma \cap \tau_v$ intersects any plate of \mathcal{T}_v . For simplicity, assume that γ_v is connected, otherwise we query with each connected component of γ_v .

If v is a leaf, we answer the intersection query naively, in $O(n_0)$ time, by inspecting all plates in \mathcal{T}_v . If $\gamma_v \subset Z(F_v)$, then we query the secondary data structure Σ_v^0 with γ_v and return the answer. So assume that $\gamma_v \not\subset Z(F_v)$. We compute all cells of $\mathbb{R}^3 \setminus Z(F_v)$ that γ_v intersects; there are at most $c_3 D$ such cells for some absolute constant $c_3 > 0$ [11]. Let τ be such a cell. We first use the secondary data structure Σ_τ^1 to detect whether γ_v intersects any plate of \mathcal{W}_τ , the set of wide plates at τ . We then recursively query at the child w_τ to detect an intersection between γ and \mathcal{T}_τ , the set of narrow plates at τ .

For intersection-detection queries, the query procedure stops as soon as an intersection between γ and \mathcal{T} is found. For reporting/counting queries, we follow the above recursive scheme, and at each node v visited by the query procedure, we either report all the plates of \mathcal{T}_v intersected by the query arc, or add up the intersection counts returned by various secondary structures and recursive calls.

Denote by $Q(m)$ the maximum query time for a subproblem involving at most m plates. Then $Q(m) = O(m)$ for $m \leq n_0$. For $m > n_0$, Propositions 2.1 and 3.2 imply that $Q(m)$ obeys the recurrence:

$$Q(m) \leq c_3 D Q(m/D^2) + O^*(m^{2/3}), \quad (2)$$

where c_3 is the constant as defined above. Again, using the fact that the recursion terminates at $m \leq n_0 = n^{1/3}$, it can be shown that $Q(m) = O^*(m^{2/3} + m^{1/2}n^{1/6}) = O^*(m^{1/2}n^{1/6})$. For $m = n$ we get $Q(n) = O^*(n^{2/3})$. Putting together everything, we obtain the following:

► **Theorem 2.2.** *A given set \mathcal{T} of n plates in \mathbb{R}^3 can be preprocessed, in expected time $O^*(n^{4/3})$, into a data structure of size $O^*(n^{4/3})$ so that an arc-intersection query amid \mathcal{T} can be answered in $O^*(n^{2/3})$ time.*

In the full version [3], we present a different technique for preprocessing a set \mathcal{T} of triangles, in expected time $O^*(n)$, into a data structure of $O^*(n)$ size that can answer arc-intersection queries in $O^*(n^{4/5})$ time. Using this data structure, we can modify our main structure Ψ , as follows: We set $n_0 = n^{5/9}$, i.e., a node v is a leaf if $n_v \leq n^{5/9}$. We construct the above structure at each leaf of Ψ . The recursion now terminates at depth i satisfying $n/D^{2i} \approx n^{5/9}$, or $D^i = n^{2/9}$. The overall query procedure is the same as above except that we use at each leaf the aforementioned improved procedure. This can be shown to yield:

► **Theorem 2.3.** *A set \mathcal{T} of n triangles in \mathbb{R}^3 can be processed, in expected time $O^*(n^{11/9})$, into a data structure of size $O^*(n^{11/9})$ so that an arc-intersection query amid the triangles of \mathcal{T} can be answered in $O^*(n^{2/3})$ time.*

2.3 Space/query-time trade-offs

As we show in the full version [3], we can improve the query time for the secondary structure on wide plates by increasing the size of the structure. Specifically, we show that a set \mathcal{W}_τ of n wide plates at some partition cell τ can be preprocessed, in expected time $O^*(n^{3/2})$, into a data structure of size $O^*(n^{3/2})$, so that the query time improves to $O^*\left(n^{\frac{2t-3}{3(t-1)}}\right)$, where $t \geq 3$ is the parametric dimension of the query arcs. We adapt our primary data

structure Ψ , as follows: (a) we now set n_0 to be a sufficiently large constant; and (b) we apply a standard primal-dual range-searching algorithm for the wide plates, instead of the primal-only approach of [20] used in the basic solution. Omitting all the details, we conclude:

► **Theorem 2.4.** *Let \mathcal{T} be a set of n plates in \mathbb{R}^3 , and let Γ be a family of arcs of parametric dimension $t \geq 3$. \mathcal{T} can be preprocessed, in expected time $O^*(n^{3/2})$, into a data structure of size $O^*(n^{3/2})$, so that an arc intersection query with an arc in Γ can be answered in $O^*(n^{\frac{2t-3}{3(t-1)}})$ time.*

3 Handling wide plates

Let \mathcal{T} be a set of n plates in \mathbb{R}^3 , Γ a family of arcs, and F a partitioning polynomial, as described in Section 2. In this section we describe the algorithm for preprocessing the set of wide plates, \mathcal{W}_τ , for each cell τ of $\mathbb{R}^3 \setminus Z(F)$, for intersection queries with arcs of Γ . Fix a cell τ . Let $\Delta \in \mathcal{W}_\tau$ be a plate that is wide at τ , and let h_Δ be the plane supporting Δ . Since Δ is wide at τ , each connected component of $\Delta \cap \tau$ is also a connected component of $h_\Delta \cap \tau$ (though some connected components of $h_\Delta \cap \tau$ may be disjoint from Δ). Roughly speaking, by a careful construction of a *cylindrical algebraic decomposition* (CAD) Ξ of F (see Section 3.2), we decompose $\Delta \cap \tau$ into $O(1)$ pseudo-trapezoids, each contained in a connected component of $\Delta \cap \tau$. We collect these pseudo-trapezoids of all wide plates at τ and cluster them into $O(1)$ families, using Ξ so that, for each family Φ , all pseudo-trapezoids within Φ can be represented by a fixed constant-complexity semi-algebraic expression (that is, predicate). Each such predicate only depends on F and on the (coefficients of the) plane supporting the pseudo-trapezoid φ (but not on the boundary of the plate containing φ). Roughly speaking, the predicate is of the form $\sigma(a, b, c, x, y)$, so that a plane $z = a_0x + b_0y + c_0$ contains a pseudo-trapezoid φ_σ so that $\sigma(a_0, b_0, c_0, x, y)$ holds precisely for those points (x, y, z) in that plane that lie in φ_σ ; see Section 3.3. This semi-algebraic representation of Φ enables us to reduce the arc-intersection query on Φ to semi-algebraic range searching in only three dimensions (Section 3.4).

3.1 An overview of cylindrical algebraic decomposition

We begin by giving a brief overview of *cylindrical algebraic decomposition* (CAD), also known as Collins’ decomposition, after its originator Collins [13]. This tool is a central ingredient of our algorithm – see Section 3.2. A detailed description can be found in [11, Chapter 5]; a possibly more accessible treatment is given in [24, Appendix A].

Given a finite set $\mathcal{F} = \{f_1, \dots, f_s\}$ of d -variate polynomials, a cylindrical algebraic decomposition induced by \mathcal{F} , denoted by $\Xi(\mathcal{F})$, is a (recursive) decomposition of \mathbb{R}^d into a finite collection of relatively open simply-shaped semi-algebraic cells of dimensions $0, \dots, d$, each homeomorphic to an open ball of the respective dimension. These cells refine the arrangement $\mathcal{A}(\mathcal{F})$ of the zero sets of the polynomials in \mathcal{F} , as described next.

Set $F = \prod_{i=1}^s f_i$. For $d = 1$, let $\alpha_1 < \alpha_2 < \dots < \alpha_t$ be the distinct real roots of F . Then $\Xi(\mathcal{F})$ is the collection of cells $\{(-\infty, \alpha_1), \{\alpha_1\}, (\alpha_1, \alpha_2), \dots, \{\alpha_t\}, (\alpha_t, +\infty)\}$. For $d > 1$, regard \mathbb{R}^d as the Cartesian product $\mathbb{R}^{d-1} \times \mathbb{R}$ and assume that x_d is a *good* direction, meaning that for any fixed $a \in \mathbb{R}^{d-1}$, $F(a, x_d)$, viewed as a polynomial in x_d , has finitely many roots.

$\Xi(\mathcal{F})$ is defined recursively from a “base” $(d - 1)$ -dimensional CAD Ξ_{d-1} , as follows. One constructs a suitable set $\mathcal{E} := \mathcal{E}(\mathcal{F})$ of polynomials in x_1, \dots, x_{d-1} (denoted by $\text{ELIM}_{X_d}(\mathcal{F})$ in [11] and by Q_b in [24]). Roughly speaking, the zero sets of polynomials in \mathcal{E} , viewed as subsets of \mathbb{R}^{d-1} , contain the projection onto \mathbb{R}^{d-1} of all intersections $Z(f_i) \cap Z(f_j)$, $1 \leq i < j \leq s$, as well as the projection of the loci in each $Z(f_i)$ where $Z(f_i)$ has a tangent

hyperplane parallel to the x_d -axis, or a singularity of some kind. The actual construction of \mathcal{E} , based on *subresultants* of \mathcal{F} , is somewhat complicated, and we refer to [11, 24] for more details.

One recursively constructs $\Xi_{d-1} = \Xi(\mathcal{E})$ in \mathbb{R}^{d-1} , which is a refinement of $\mathcal{A}(\mathcal{E})$ into topologically trivial open cells of dimensions $0, 1, \dots, d-1$. For each cell $\tau \in \Xi_{d-1}$, the sign of each polynomial in \mathcal{E} is constant (zero, positive, or negative) and the (finite) number of distinct real x_d -roots of $F(\mathbf{x}, x_d)$ is the same for all $\mathbf{x} \in \tau$. $\Xi(\mathcal{F})$ is then defined in terms of Ξ_{d-1} , as follows. Fix a cell $\tau \in \Xi_{d-1}$. Let $\tau \times \mathbb{R}$ denote the *cylinder* over τ . There is an integer $t \geq 0$ such that for all $\mathbf{x} \in \tau$, there are exactly t distinct real roots $\psi_1(\mathbf{x}) < \dots < \psi_t(\mathbf{x})$ of $F(\mathbf{x}, x_d)$ (regarded as a polynomial in x_d), and these roots are algebraic functions that vary continuously with $\mathbf{x} \in \tau$. Let ψ_0, ψ_{t+1} denote the constant functions $-\infty$ and $+\infty$, respectively. Then we create the following cells that decompose the cylinder over τ :

- $\sigma = \{(\mathbf{x}, \psi_i(\mathbf{x})) \mid \mathbf{x} \in \tau\}$, for $i = 1, \dots, t$; σ is a section of the graph of ψ_i over τ , and
- $\sigma = \{(\mathbf{x}, y) \mid \mathbf{x} \in \tau, y \in (\psi_i(\mathbf{x}), \psi_{i+1}(\mathbf{x}))\}$, for $0 \leq i \leq t$; σ is a portion ('layer') of the cylinder $\tau \times \mathbb{R}$ between the two consecutive graphs ψ_i, ψ_{i+1} .

The main property of Ξ is that, for each cell $\tau \in \Xi$, the sign of each polynomial in \mathcal{F} is constant for all $\mathbf{x} \in \tau$. Omitting all further details (for which see [11, 13, 24]), we have the following lemma:

► **Lemma 3.1.** *Let $\mathcal{F} = \{f_1, \dots, f_s\}$ be a set of s d -variate polynomials of degree at most D each. Then, assuming that all coordinates are good directions, $\Xi(\mathcal{F})$ consists of $O(Ds)^{2^d}$ cells, and each cell can be represented semi-algebraically by $O(D)^{2^d}$ polynomials of degree at most $O(D)^{2^{d-1}}$. $\Xi(\mathcal{F})$ can be constructed in time $(Ds)^{2^{O(d)}}$ in a suitable standard model of algebraic computation.*

3.2 Constructing a CAD of the partitioning polynomial

Let \mathbb{E}_3 denote the space of all planes in \mathbb{R}^3 . More precisely, \mathbb{E}_3 is the (dual) three-dimensional space where each plane $h: z = ax + by + c$ is mapped to the point (a, b, c) . For $(a_0, b_0, c_0) \in \mathbb{E}_3$, we use $h(a_0, b_0, c_0)$ to denote the plane $z = a_0x + b_0y + c_0$. We consider the five-dimensional parametric space $\mathbb{E} := \mathbb{E}_3 \times \mathbb{R}^2$ with coordinates (a, b, c, x, y) . We construct in \mathbb{E} a CAD of the single 5-variate polynomial $F(x, y, ax + by + c)$. We use a generic choice of coordinates to ensure that all the axes of the coordinate frame are in good directions for the construction of the CAD, coming up next. Such a generic choice of coordinates also allows us to assume that none of the input plates lies in a vertical plane.

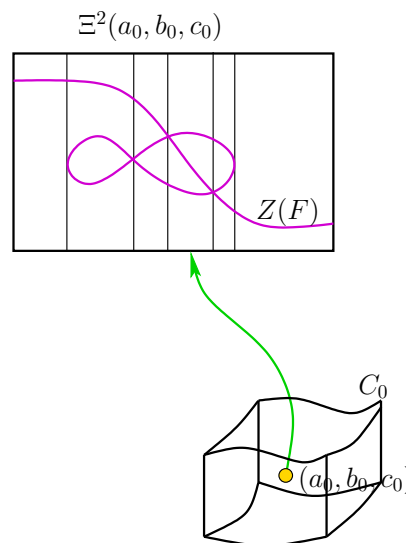
The construction of the CAD recursively eliminates the variables in the order y, x, c, b, a . That is, unfolding the recursive definition given in Section 3.1, each cell of the CAD is given by a sequence of equalities or inequalities (one from each row) of the form:

$$\begin{array}{lll}
 a = a_0 & \text{or} & a_0^- < a < a_0^+ \\
 b = f_1(a) & \text{or} & f_1^-(a) < b < f_1^+(a) \\
 c = f_2(a, b) & \text{or} & f_2^-(a, b) < c < f_2^+(a, b) \\
 x = f_3(a, b, c) & \text{or} & f_3^-(a, b, c) < x < f_3^+(a, b, c) \\
 y = f_4(a, b, c; x) & \text{or} & f_4^-(a, b, c; x) < y < f_4^+(a, b, c; x),
 \end{array} \tag{3}$$

where a_0, a_0^-, a_0^+ are real parameters, and $f_1, f_1^-, f_1^+, \dots, f_4, f_4^-, f_4^+$ are constant-degree continuous algebraic functions (any of which can be $\pm\infty$), so that, whenever we have an inequality involving two reals or two functions, we then have $a_0^- < a_0^+$, and/or $f_1^-(a) < f_1^+(a)$, $f_2^-(a, b) < f_2^+(a, b)$, $f_3^-(a, b, c) < f_3^+(a, b, c)$, and $f_4^-(a, b, c; x) < f_4^+(a, b, c; x)$, over the cell defined by the preceding set of equalities and inequalities in (3).

Let $\Xi_5 = \Xi_5(F)$ denote the five-dimensional CAD just defined, and let Ξ_3 denote the projection of Ξ_5 onto \mathbb{E}_3 , which we refer to as the *base* of Ξ_5 and which itself is a CAD of a suitable set of polynomials. Each base cell of Ξ_3 is given by a set of equalities and inequalities from the first three rows of (3), one per row. For a point $(a_0, b_0, c_0) \in \mathbb{E}_3$, let $\Xi^2(a_0, b_0, c_0)$ denote the decomposition in the xy -subspace that is induced by Ξ_5 over (a_0, b_0, c_0) . This is the decomposition of the xy -plane into pseudo-trapezoids, each of which is given by equalities and/or inequalities from the last two rows of (3), with $a = a_0, b = b_0, c = c_0$. We refer to $\Xi^2(a_0, b_0, c_0)$ as the two-dimensional *fiber* of Ξ_5 over (a_0, b_0, c_0) . As a matter of fact, and this is the main rationale for the CAD construction, $\Xi^2(a_0, b_0, c_0)$ can be identified with the xy -projection of a refinement of the partition induced by $Z(F)$ in the plane $h(a_0, b_0, c_0)$. That is, each 2-cell of this two-dimensional fiber of Ξ_5 is contained in the projection of a single connected component of $h(a_0, b_0, c_0) \setminus Z(F)$, and each 0-cell, as well as each 1-cell that is not y -vertical, of the fiber is contained in the projection of a portion of $Z(F) \cap h(a_0, b_0, c_0)$. See Figure 1 for an illustration.

The topology of the partition induced by $Z(F)$ in $h(a_0, b_0, c_0)$ does not change as long as (a_0, b_0, c_0) stays in the same cell C_0 of Ξ_3 , and changes in the topology occur only when we cross between cells of Ξ_3 . In particular, each cell C of Ξ_5 can be associated with a fixed cell of $\mathbb{R}^3 \setminus Z(F)$, denoted as τ_C , such that for all points (a_0, b_0, c_0) in the base cell $C^\downarrow \subset \mathbb{E}_3$ of C , which is the projection of C onto \mathbb{E}_3 , the two-dimensional portion C^2 of the fiber $\Xi^2(a_0, b_0, c_0)$ for which $\{(a_0, b_0, c_0)\} \times C^2 \subseteq C$ is the xy -projection of a pseudo-trapezoid of a connected component of $h(a_0, b_0, c_0) \cap \tau_C$. This property will be useful in constructing the data structure to answer arc-intersection queries amid the wide plates at τ .



■ **Figure 1** An illustration of the CAD construction. C_0 is a three-dimensional cell of Ξ_3 . For a point $(a_0, b_0, c_0) \in C_0$, its two-dimensional fiber $\Xi^2(a_0, b_0, c_0)$ is shown. Formally, the purple curve is the xy -projection of $Z(F) \cap h(a_0, b_0, c_0)$.

3.3 Decomposing wide plates into pseudo-trapezoids

We are now ready to describe how to decompose each plate $\Delta \in \mathcal{W}_\tau$, for each cell τ of $\mathbb{R}^3 \setminus Z(F)$, into pseudo-trapezoids, and how to cluster the resulting pseudo-trapezoids. Let $\Delta \in \mathcal{T}$ be a plate, let h_Δ be the plane supporting Δ , and let $\Delta^* = (a_0, b_0, c_0)$ be the point in

the abc -subspace \mathbb{E}_3 dual to h_Δ . We locate (in constant time, by brute force) the cell C_0 of Ξ_3 (in \mathbb{E}_3) that contains Δ^* . Let φ be a cell of $\Xi^2(\Delta^*)$, let $\varphi^\uparrow = \{(x, y, a_0x + b_0y + c_0) \mid (x, y) \in \varphi\}$ be the lifting of φ onto h_Δ , and let C be the cell of Ξ_5 that contains $\{\Delta^*\} \times \varphi$. We determine whether φ^\uparrow is fully contained in Δ , lies fully outside Δ , or intersects $\partial\Delta$. We keep φ only if φ^\uparrow is contained in Δ , and associate φ^\uparrow , as well as the plate Δ , with C . (In general, Δ is associated with many cells C , one for each cell φ of $\Xi^2(\Delta^*)$ whose lifting is contained in Δ .) In this case, we use Δ_C to denote the pseudo-trapezoid φ^\uparrow , which is uniquely determined by Δ and C and which lies in a connected component of $\Delta \cap \tau_C$. For a cell $C \in \Xi_5$, let $\mathcal{T}_C \subseteq \mathcal{T}$ be the subset of plates that are associated with C , and let $\Phi_C = \{\Delta_C \mid \Delta \in \mathcal{T}_C\}$ be the subset of pseudo-trapezoids associated with C . Finally, for a plate $\Delta \in \mathcal{T}$, let Ξ_Δ be the set of all cells of Ξ_5 with which Δ is associated. Again, see Figure 1 for an illustration.

The advantage of this approach is that for each plate $\Delta \in \mathcal{T}$, the set $\Delta^\parallel := \{\Delta_C \mid C \in \Xi_\Delta\}$ is a refinement into pseudo-trapezoids of those cells of $h_\Delta \setminus Z(F)$, referred to as *inner* cells, that lie fully inside Δ . Furthermore, the set Ξ_Δ provides an operational “labeling” scheme for the pseudo-trapezoids in Δ^\parallel – the pseudo-trapezoid Δ_C is labeled with C , or rather with the semi-algebraic representation that it inherits from C . That is, each such pseudo-trapezoid φ^\uparrow on the plate Δ , with the point Δ^* belongs to some base cell C_0 of Ξ_3 , is represented by equalities and inequalities of the form

$$x = f_3(\Delta^*) \quad \text{or} \quad f_3^-(\Delta^*) < x < f_3^+(\Delta^*) \quad \text{and} \quad y = f_4(\Delta^*) \quad \text{or} \quad f_4^-(\Delta^*) < y < f_4^+(\Delta^*),$$

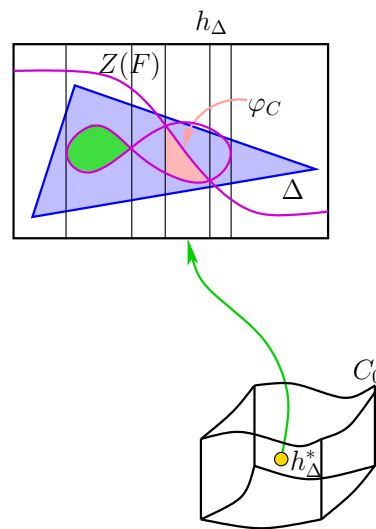
where $f_3, f_3^-, f_3^+, f_4, f_4^-, f_4^+$ are constant-degree continuous algebraic functions over the corresponding domains, as in (3). This is a simple semi-algebraic representation, of constant complexity, of the xy -projection φ of φ^\uparrow , which *does not explicitly depend on* Δ (but only on its plane h_Δ). Moreover, this representation is fixed for all plates Δ for which the points Δ^* lie in the same cell of Ξ_3 , and is therefore also independent of h_Δ ,⁵ as long as Δ^* belongs to that cell. See Figure 2 for an illustration. This constant-size “labeling” is used for clustering the pseudo-trapezoids into which the inner cells of h_Δ , for $\Delta \in \mathcal{T}$, are partitioned. Namely, we put all pseudo-trapezoids labeled with the same cell C of Ξ_5 into one cluster, and $\{\Phi_C \mid C \in \Xi_5\}$ is the desired clustering of the pseudo-trapezoids.

3.4 Reduction to semi-algebraic range searching

Fix a cell C of Ξ_5 . For an arc $\gamma \in \Gamma$, contained in the cell τ_C of $\mathbb{R}^3 \setminus Z(F)$, we wish to answer an arc-intersection query on Δ_C with γ . To this end, we define the predicate $\Pi_C: \Gamma \times \mathbb{E}_3 \rightarrow \{0, 1\}$ so that $\Pi_C(\gamma; a, b, c)$ is 1 if and only if γ crosses $h(a, b, c)$ at a point (x, y, z) such that (x, y) belongs to C (that is, $(a, b, c, x, y) \in C$), and (x, y, z) lies in τ_C . It is easy to verify that $\Pi_C(\gamma; a, b, c)$ is a semi-algebraic predicate of constant complexity (that depends on D and t , the parametric dimension of arcs in Γ). We now define the semi-algebraic range $Q_{C, \gamma} := \{(a, b, c) \mid \Pi_C(\gamma; a, b, c) = 1\}$, which is of constant complexity too. By construction, γ crosses Δ_C if and only if the point $\Delta^* \in Q_{C, \gamma}$. Set $\mathcal{T}_C^* := \{\Delta^* \mid \Delta \in \mathcal{T}_C\}$.

For each cell $C \in \Xi_5$, we preprocess $\mathcal{T}_C^* \subset \mathbb{E}_3$, in $O(|\mathcal{T}_C| \log n)$ expected time, into a data structure Σ_C of size $O(|\mathcal{T}_C|)$, using the range-searching mechanism of Matoušek and Patáková [20] (see also [8]). For a query range $Q_{C, \gamma}$, the range query on \mathcal{T}_C^* can be answered in $O^*(|\mathcal{T}_C|^{2/3})$ time.

⁵ More precisely, its dependence on h_Δ is only in terms of the coefficients (a, b, c) of h_Δ that are substituted in the fixed semi-algebraic predicate given above.



■ **Figure 2** The labeling scheme provided by the CAD (the plate depicted in this figure is a triangle). The cell C labels, by an explicit semi-algebraic expression, the highlighted inner pseudo-trapezoidal subcell φ_C within the plate Δ . Another inner subcell, with a different label, in a different partition cell τ , is also highlighted.

Finally, for a cell τ of $\mathbb{R}^3 \setminus Z(F)$, let $\Xi_\tau = \{C \in \Xi_5 \mid \tau_C = \tau\}$ be the set of all CAD cells associated with τ . We store the structures Σ_C , for all $C \in \Xi_\tau$, at τ as the secondary structure Σ_τ^1 . To test whether an arc $\gamma \in \Gamma$, which lies inside τ , intersects a plate of \mathcal{W}_τ , we query each of the structures Σ_C stored at τ with $Q_{C,\gamma}$ and return yes if any of them returns yes. Putting everything together, we obtain the following:

► **Proposition 3.2.** *A set \mathcal{W} of n wide plates at some cell τ can be preprocessed into a data structure of size $O^*(n)$, in $O^*(n)$ expected time, so that an arc-intersection query, for intersections within τ , can be answered in $O^*(n^{2/3})$ time.*

This at last completes the analysis for the wide plates, which implies the main result of this paper.

References

- 1 P. K. Agarwal. Ray shooting and other applications of spanning trees with low stabbing number. *SIAM J. Comput.*, 21(3):540–570, 1992.
- 2 P. K. Agarwal. Simplex range searching and its variants: A review. In *Journey through Discrete Mathematics: A Tribute to Jiří Matoušek*, pages 1–30. Springer Verlag, Berlin-Heidelberg, 2017.
- 3 P. K. Agarwal, B. Aronov, E. Ezra, M. J. Katz, and M. Sharir. Intersection queries for flat semi-algebraic objects in three dimensions and related problems. [arXiv:2203.10241](https://arxiv.org/abs/2203.10241).
- 4 P. K. Agarwal, B. Aronov, E. Ezra, and J. Zahl. An efficient algorithm for generalized polynomial partitioning and its applications. *SIAM J. Comput.*, 50:760–787, 2021. Also in *Proc. Sympos. on Computational Geometry (SoCG)*, 2019, 5:1–5:14. Also in [arXiv:1812.10269](https://arxiv.org/abs/1812.10269).
- 5 P. K. Agarwal, B. Aronov, and S. Suri. Stabbing triangulations by lines in 3D. In *Proc. 11th Annu. Sympos. on Computational Geometry*, pages 267–276, 1995.
- 6 P. K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discret. Comput. Geom.*, 11:393–418, 1994.

- 7 P. K. Agarwal and J. Matoušek. Ray shooting and parameric search. *SIAM J. Comput.*, 22:794–806, 1993.
- 8 P. K. Agarwal, J. Matoušek, and M. Sharir. On range searching with semialgebraic sets II. *SIAM J. Comput.*, 42:2039–2062, 2013. Also in [arXiv:1208.3384](#).
- 9 P. K. Agarwal and Micha Sharir. Ray shooting amidst convex polyhedra and polyhedral terrains in three dimensions. *SIAM J. Comput.*, 25(1):100–116, 1996.
- 10 B. Aronov and S. Fortune. Approximating minimum-weight triangulations in three dimensions. *Discrete Comput. Geom.*, 21(4):527–549, 1999.
- 11 S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*. Algorithms and Computation in Mathematics 10. Springer-Verlag, Berlin, 2003.
- 12 B. Chazelle. Fast searching in a real algebraic manifold with applications to geometric complexity. In *Colloquium on Trees in Algebra and Programming*, pages 145–156, 1985.
- 13 G. E. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conf. Automata Theory and Formal Languages*, volume 33 of *LNCS*, pages 134–183. Springer, 1975.
- 14 M. de Berg, D. Halperin, M. H. Overmars, J. Snoeyink, and M. J. van Kreveld. Efficient ray shooting and hidden surface removal. *Algorithmica*, 12(1):30–53, 1994.
- 15 D. P. Dobkin and D. G. Kirkpatrick. Fast detection of polyhedral intersection. *Theor. Comput. Sci.*, 27:241–253, 1983.
- 16 E. Ezra and M. Sharir. On ray shooting for triangles in 3-space and related problems. *SIAM J. Comput.*, in press. Also in *Proc. 37th Sympos. on Computational Geometry (2021)*, 34:1–34:15. Also in [arXiv:2102.07310](#).
- 17 L. Guth. Polynomial partitioning for a set of varieties. *Math. Proc. Camb. Phil. Soc.*, 159:459–469, 2015. Also in [arXiv:1410.8871](#).
- 18 L. Guth and N. H. Katz. On the Erdős distinct distances problem in the plane. *Annals Math.*, 181:155–190, 2015. Also in [arXiv:1011.4105](#).
- 19 J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18(3):403–431, 1995.
- 20 J. Matoušek and Z. Patáková. Multilevel polynomial partitions and simplified range searching. *Discrete Comput. Geom.*, 54:22–41, 2015.
- 21 S. Mohabab and M. Sharir. Ray shooting amidst spheres in 3 dimensions and related problems. *SIAM J. Comput.*, 26:654–674, 1997.
- 22 M. Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9:471–494, 1993.
- 23 M. Pellegrini. Ray shooting and lines in space. In *Handbook on Discrete and Computational Geometry*, chapter 41, pages 1093–1112. CRC Press, Boca Raton, Florida, 3rd edition, 2017.
- 24 J. T. Schwartz and M. Sharir. On the Piano Movers’ problem: II. General techniques for computing topological properties of real algebraic manifolds. *Advances in Appl. Math.*, 4:298–351, 1983.
- 25 M. Sharir and H. Shaul. Ray shooting and stone throwing with near-linear storage. *Comput. Geom. Theory Appl.*, 30:239–252, 2005.
- 26 A. C. Yao and F. F. Yao. A general approach to d-dimensional geometric queries (extended abstract). In *Proc. 17th Annu. ACM Symp. Theory of Computing*, pages 163–168, 1985.

Twisted Ways to Find Plane Structures in Simple Drawings of Complete Graphs

Oswin Aichholzer ✉ 

Institute of Software Technology, Technische Universität Graz, Austria

Alfredo García ✉ 

Departamento de Métodos Estadísticos and IUMA, University of Zaragoza, Spain

Javier Tejel ✉ 

Departamento de Métodos Estadísticos and IUMA, University of Zaragoza, Spain

Birgit Vogtenhuber ✉ 

Institute of Software Technology, Technische Universität Graz, Austria

Alexandra Weinberger ✉ 

Institute of Software Technology, Technische Universität Graz, Austria

Abstract

Simple drawings are drawings of graphs in which the edges are Jordan arcs and each pair of edges share at most one point (a proper crossing or a common endpoint). We introduce a special kind of simple drawings that we call generalized twisted drawings. A simple drawing is generalized twisted if there is a point O such that every ray emanating from O crosses every edge of the drawing at most once and there is a ray emanating from O which crosses every edge exactly once.

Via this new class of simple drawings, we show that every simple drawing of the complete graph with n vertices contains $\Omega(n^{\frac{1}{2}})$ pairwise disjoint edges and a plane path of length $\Omega(\frac{\log n}{\log \log n})$. Both results improve over previously known best lower bounds. On the way we show several structural results about and properties of generalized twisted drawings. We further present different characterizations of generalized twisted drawings, which might be of independent interest.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory

Keywords and phrases Simple drawings, simple topological graphs, disjoint edges, plane matching, plane path

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.5

Related Version Some results of this work have been presented at the Computational Geometry: Young Researchers Forum in 2021 [3] and at the Encuentros de Geometría Computacional 2021 [4].

Extended Version: <https://arxiv.org/abs/2203.06143v1>

Funding *Oswin Aichholzer:* Partially supported by the Austrian Science Fund (FWF): W1230 and by H2020-MSCA-RISE project 734922 – CONNECT.

Alfredo García: Supported by H2020-MSCA-RISE project 734922 – CONNECT and Gobierno de Aragón project E41-17R.

Javier Tejel: Supported by H2020-MSCA-RISE project 734922 – CONNECT, Gobierno de Aragón project E41-17R and project PID2019-104129GB-I00 / AEI / 10.13039/501100011033 of the Spanish Ministry of Science and Innovation.

Birgit Vogtenhuber: Partially supported by Austrian Science Fund within the collaborative DACH project *Arrangements and Drawings* as FWF project I 3340-N35 and by H2020-MSCA-RISE project 734922 – CONNECT.

Alexandra Weinberger: Supported by the Austrian Science Fund (FWF): W1230 and by H2020-MSCA-RISE project 734922 – CONNECT.



© Oswin Aichholzer, Alfredo García, Javier Tejel, Birgit Vogtenhuber, and Alexandra Weinberger;

licensed under Creative Commons License CC-BY 4.0

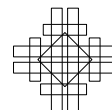
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 5; pp. 5:1–5:18

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Simple drawings are drawings of graphs in the plane such that vertices are distinct points in the plane, edges are Jordan arcs connecting their endpoints, and edges intersect at most once either in a proper crossing or in a shared endpoint. The edges and vertices of a drawing partition the plane (or, more exactly, the plane minus the drawing) into regions, which are called the *cells* of the drawing. If a simple drawing is plane (that is, crossing-free), then its cells are classically called *faces*.

In the past decades, there has been significant interest in simple drawings. Questions about plane subdrawings of simple drawings of the complete graph on n vertices, K_n , have attracted particularly close attention.

Rafla [18] conjectured that every simple drawing of K_n contains a plane Hamiltonian cycle. The conjecture has been shown to hold for $n \leq 9$ [1], as well as for several special classes of simple drawings, like straight-line, monotone, and cylindrical drawings, but remains open in general. If Rafla's conjecture is true, then this would immediately imply that every simple drawing of the complete graph contains a plane perfect matching. However, to-date even the existence of such a matching is still unknown.

Ruiz-Vargas [20] showed in 2017 that every simple drawing of K_n contains $\Omega(n^{\frac{1}{2}-\varepsilon})$ pairwise disjoint edges for any $\varepsilon > 0$, which improved over a series of previous results: $\Omega((\log n)^{\frac{1}{6}})$ in 2003 [15], $\Omega(\frac{\log n}{\log \log n})$ in 2005 [16], $\Omega((\log n)^{1+\varepsilon})$ in 2009 [9], and $\Omega(n^{\frac{1}{3}})$ in 2013 and 2014 [10, 12, 21].

Pach, Solymosi, and Tóth [15] showed that every simple drawing of K_n contains a subdrawing of $K_{c \log^{\frac{1}{8}} n}$, for some constant c , that is either *convex* or *twisted*¹. They further showed that every simple drawing of K_n contains a plane subdrawing isomorphic to any fixed tree with up to $c \log^{\frac{1}{6}} n$ vertices, for some constant c . This implies that every simple drawing of K_n contains a plane path of length $\Omega((\log n)^{\frac{1}{6}})$, which has been the best lower bound known prior to this paper.

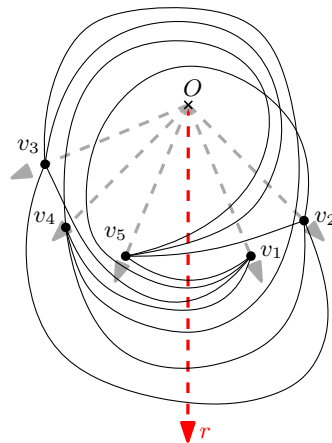
Concerning general plane substructures, it follows from a result of Ruiz-Vargas [20] that every simple drawing of K_n contains a plane subdrawing with at least $2n - 3$ edges. Further, García, Pilz, and Tejel [13] showed that every maximal plane subdrawing of a simple drawing of K_n is biconnected. Note that, in contrast to straight-line drawings, simple drawings of K_n in general do not contain triangulations, that is, plane subdrawings where all faces (except at most one) are 3-cycles.

In this paper, we introduce a new family of simple drawings, which we call *generalized twisted* drawings. The name stems from the fact that one can show that any twisted drawing is weakly isomorphic to a generalized twisted drawing (but not every generalized twisted drawing is weakly isomorphic to a twisted drawing). It follows, that for any n there exists a generalized twisted drawing. Two drawings D and D' are *weakly isomorphic* if there is a bijection between the vertices and edges of D and D' such that a pair of edges in D crosses exactly when the corresponding pair of edges in D' crosses.

► **Definition 1.** *A simple drawing D is **c-monotone** (short for circularly monotone) if there is a point O such that any ray emanating from O intersects any edge of D at most once.*

*A simple drawing D of K_n is **generalized twisted** if there is a point O such that D is **c-monotone** with respect to O and there exists a ray r emanating from O that intersects every edge of D .*

¹ In their definition for simple drawings, *convex* means that there is a labeling of the vertices to v_1, v_2, \dots, v_n such that (v_i, v_j) ($i < j$) crosses (v_k, v_l) ($k < l$) if and only if $i < k < j < l$ or $k < i < l < j$, and *twisted* means that there is a labeling of the vertices to v_1, v_2, \dots, v_n such that (v_i, v_j) ($i < j$) crosses (v_k, v_l) ($k < l$) if and only if $i < k < l < j$ or $k < i < j < l$.



■ **Figure 1** A generalized twisted drawing of K_5 . All edges cross the (red) ray r .

We label the vertices of c -monotone drawings v_1, \dots, v_n in counterclockwise order around O . In generalized twisted drawings, they are labeled such that the ray r emerges from O between the ray to v_1 and the one to v_n . Figure 1 shows an example of a generalized twisted drawing of K_5 .

Generalized twisted drawings turn out to have quite surprising structural properties. We show some crossing properties of generalized twisted drawings in Section 2 and with that also prove that they always contain plane Hamiltonian paths (Theorem 3). This result is an essential ingredient for showing that any simple drawing of K_n contains $\Omega(\sqrt{n})$ pairwise disjoint edges (Theorem 9 in Section 3), as well as a plane path of length $\Omega(\frac{\log n}{\log \log n})$ (Theorem 10 in Section 4). In Section 5, we present different characterizations of generalized twisted drawings that are of independent interest. We conclude with an outlook on further work and open problems in Section 6. *Full proofs are available at [arXiv:2203.06143](https://arxiv.org/abs/2203.06143).*

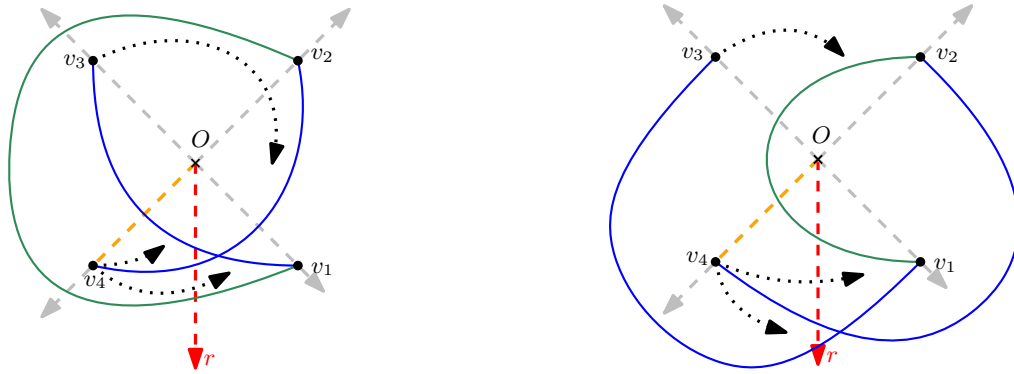
2 Twisted preliminaries

In this section, we show some properties of generalized twisted drawings, which will be used in the following sections.

► **Lemma 2.** *Let D be a generalized twisted drawing of K_4 , with vertices $\{v_1, v_2, v_3, v_4\}$ labeled counterclockwise around O . Then the edges v_1v_3 and v_2v_4 do not cross.*

Proof Sketch. Assume, for a contradiction, that the edge v_1v_3 crosses the edge v_2v_4 . There are (up to strong isomorphism) two possibilities to draw the crossing edges v_1v_3 and v_2v_4 , depending on whether v_1v_3 crosses the (straight-line) segment from O to v_4 or not; cf. Figure 2. In both cases, there is only one way to draw v_1v_2 such that the drawing stays generalized twisted, yielding two regions bounded by all drawn edges (v_1v_3, v_2v_4, v_1v_2) . The vertices v_3 and v_4 lie in the same region. It is well-known that every simple drawing of K_4 has at most one crossing. Thus, the edge v_3v_4 cannot leave this region. However, it is impossible to draw v_3v_4 without leaving the region such that it is c -monotone and crosses the ray r (see the dotted arrows in Figure 2 for necessary emanating directions of v_3v_4). ◀

Using the crossing property of Lemma 2, it follows directly that generalized twisted drawings always contain plane Hamiltonian paths.



■ **Figure 2** The two possibilities to draw v_1v_3 and v_2v_4 crossing and generalized twisted.

► **Theorem 3.** *Every generalized twisted drawing of K_n contains a plane Hamiltonian path.*

Proof of Theorem 3. Let D be a generalized twisted drawing of K_n . Consider the Hamiltonian path $v_1, v_{\lceil \frac{n}{2} \rceil + 1}, v_2, v_{\lceil \frac{n}{2} \rceil + 2}, v_3, \dots, v_{\lceil \frac{n}{2} \rceil - 1}, v_n, v_{\lfloor \frac{n}{2} \rfloor}$ if n is odd or the Hamiltonian path $v_1, v_{\lceil \frac{n}{2} \rceil + 1}, v_2, v_{\lceil \frac{n}{2} \rceil + 2}, v_3, \dots, v_{n-1}, v_{\lceil \frac{n}{2} \rceil}, v_n$ if n is even. See for example the Hamiltonian path v_1, v_4, v_2, v_5, v_3 in Figure 1. Take any pair of edges (v_i, v_j) and (v_k, v_l) of the path, where we can assume without loss of generality that $i < j$ and $k < l$. If the two edges share an endpoint, they are adjacent and do not cross. Otherwise, if they do not share an endpoint, either $i < k < j < l$ or $k < i < l < j$ by definition of the path. In any of the two cases, (v_i, v_j) and (v_k, v_l) cannot cross by Lemma 2. Therefore, no pair of edges cross, and the Hamiltonian path is plane. ◀

Analogous to the proof of Theorem 3, one can argue that in every generalized twisted drawing of K_n with n odd, the Hamiltonian cycle $v_1, v_{\lceil \frac{n}{2} \rceil + 1}, v_2, v_{\lceil \frac{n}{2} \rceil + 2}, \dots, v_{\lceil \frac{n}{2} \rceil - 1}, v_n, v_{\lfloor \frac{n}{2} \rfloor}, v_1$ is plane. We strongly conjecture that every generalized twisted drawing of K_n contains a plane Hamiltonian cycle, but its structure for even n is still an open problem.

Theorem 3 will be used heavily in the next two sections. Further, the following statement, which has been implicitly shown in [10] and [12], will be used in all remaining sections.

► **Lemma 4.** *Let D be a simple drawing of a complete graph containing a subdrawing D' , which is a plane drawing of $K_{2,n}$. Let $A = \{a_1, a_2, \dots, a_n\}$ and $B = \{b_1, b_2\}$ be the sides of the bipartition of D' . Let D_A be the subdrawing of D induced by the vertices of A . Then D_A is weakly isomorphic to a c -monotone drawing. Moreover, if all edges in D_A cross the edge b_1b_2 , then D_A is weakly isomorphic to a generalized twisted drawing.*

3 Disjoint edges in simple drawings

In this section, we show that every simple drawing of K_n contains at least $\lfloor \sqrt{\frac{n}{48}} \rfloor$ pairwise disjoint edges, improving the previously known best bound of $\Omega(n^{\frac{1}{2}-\varepsilon})$, for any $\varepsilon > 0$, by Ruiz-Vargas [20]. In addition to the properties of generalized twisted drawings from Section 2, we use the following theorems and observations to prove this new lower bound.

► **Theorem 5 ([13]).** *For $n \geq 3$, every maximal plane subdrawing of any simple drawing of K_n is biconnected.*

The following theorem is a direct consequence of Corollary 5 in [19].

► **Theorem 6.** *Let D be a simple drawing of K_n with $n \geq 3$. Let H be a connected plane subdrawing of D containing at least two vertices, and let v be a vertex in $D \setminus H$. Then D contains two edges incident to v that connect v with H and do not cross any edges of H .*

► **Observation 7.** *For any $n \geq 3$, the number of edges in a planar graph with n vertices is at most $3n - 6$.*

A drawing is *outerplane* if it is plane, and all vertices lie on the unbounded face of the drawing. A graph is *outerplanar* if it can be drawn outerplane. Outerplanar graphs have a smaller upper bound on their number of edges than planar graphs.

► **Observation 8.** *For any $n \geq 3$, the number of edges in an outerplanar graph with n vertices is at most $2n - 3$.*

► **Theorem 9.** *Every simple drawing of K_n contains at least $\lfloor \sqrt{\frac{n}{48}} \rfloor$ pairwise disjoint edges.*

Proof. Let D be a simple drawing of K_n , and let M be a maximal plane matching of D . If $m := |M| \geq \sqrt{\frac{n}{48}}$, then Theorem 9 holds. So assume that $|M| < \sqrt{\frac{n}{48}}$. We will show how to find another plane matching, whose size is at least $\lfloor \sqrt{\frac{n}{48}} \rfloor$.

The overall idea is the following: Let H be a maximal plane subdrawing of D whose vertex set is exactly the vertices matched in M and that contains M . We will find a face f in H that contains much more unmatched vertices inside than matched vertices on its boundary. Then we will show that there exists a subset of the vertices inside that face, which induces a subdrawing of D that is weakly isomorphic to a generalized twisted drawing and contains enough vertices to guarantee the desired size of the plane matching.

We start towards finding the face f . By Theorem 5, H is biconnected. Thus, H partitions the plane into faces, where the boundary of each face is a simple cycle. Note that the vertices of H are exactly the vertices that are matched in M , and the vertices inside faces are the vertices that are unmatched in M . Let U be the set of vertices of D that are not matched by any edge of M . We denote the set of vertices of U inside a face f_i by $U(f_i)$, the number of vertices in $U(f_i)$ by $u(f_i)$, and the number of vertices on the boundary of the face f_i by $|f_i|$.

We next show that there exists a face f of H such that $u(f) \geq \frac{\sqrt{48n}}{12}|f|$. Assume for a contradiction that for every face f_i it holds that

$$u(f_i) < \frac{\sqrt{48n}}{12}|f_i|.$$

There are exactly $n - 2m$ unmatched vertices. As every unmatched vertex is in the interior of a face of H (that might be the unbounded face), we can count the unmatched vertices by summing over the number of vertices in each face (including the unbounded face). Thus,

$$n - 2m \leq \sum_{f_i} u(f_i) < \frac{\sqrt{48n}}{12} \sum_{f_i} |f_i|. \tag{1}$$

The number of edges in H is $\frac{1}{2} \sum_{f_i} |f_i|$. Since H is plane, we can use Observation 7 to bound the number of edges of H by $3n' - 6$, where n' is the number of vertices in H . As the vertices of H are exactly the matched vertices, their number is $n' = 2m$. Hence,

$$\sum_{f_i} |f_i| \leq 6 \cdot 2m - 12.$$

From $m < \sqrt{\frac{n}{48}}$ it follows that

$$\sum_{f_i} |f_i| < 12\sqrt{\frac{n}{48}} - 12 \tag{2}$$

and

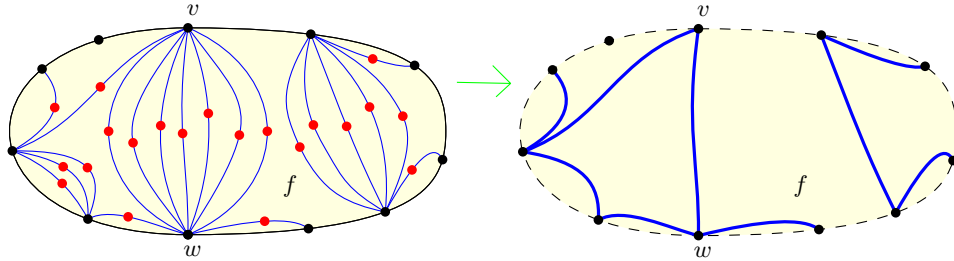
$$n - 2\sqrt{\frac{n}{48}} < n - 2m. \tag{3}$$

Putting equations (1) to (3) together we obtain that

$$n - 2\sqrt{\frac{n}{48}} < \frac{\sqrt{48n}}{12}(12\sqrt{\frac{n}{48}} - 12) = n - \sqrt{48n}.$$

However, this inequality cannot be fulfilled by any $n \geq 0$. Thus, there exists at least one face f_i with $u(f_i) \geq \frac{\sqrt{48n}}{12}|f_i|$. We call that face f . (If there are several such faces, we take an arbitrary one of them and call it f .)

As a next step, we will find two vertices on the boundary of f to which many vertices inside f are connected via edges that do not cross each other or H . From f and the set $U(f)$, we construct a plane subdrawing H' as follows; cf. Figure 3 (left). We add the vertices and edges on the boundary of f . Then we iteratively add all the vertices in $U(f)$, where for each added vertex v we also add two edges of D incident to v such that the resulting drawing stays plane. Two such edges exist by Theorem 6. Since the matching M is maximal, any edges between two unmatched vertices must cross at least one edge of M and thus must cross the boundary of f . Hence, no edge in H' can connect two vertices of $U(f)$ (as they are unmatched). Consequently, every vertex in $U(f)$ is connected in H' to exactly two vertices that both lie on the boundary of f .



■ **Figure 3** Left: The face f in H containing the plane drawing H' (blue lines) inside. Right: We can obtain an outerplane drawing from H' by interpreting bundles of edge pairs incident to the same black vertices as plane edges.

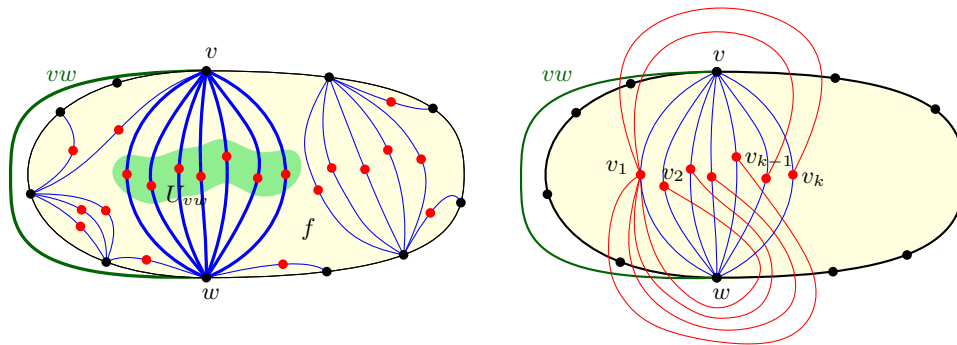
We consider the edges in H' that connect a vertex in $U(f)$ as a pair of edges. Every edge in such a pair is contained in exactly one pair, since it is incident to exactly one unmatched vertex. Thus, we can see every such pair of edges as one *long edge* incident to two vertices on the boundary of f . If several of those long edges have the same endpoints, we call them a bundle of edges; see Figure 3 (right).

From the long edges, we can define a graph G' as follows. The vertices of G' are the vertices of D that lie on the boundary of f . Two vertices u and v are connected in G' if there is at least one long edge in H' that connects them. By the definition of long edges, G' is outerplanar (as can be observed in Figure 3 (right)). Note that every unmatched vertex in

$U(f)$ defines a long edge, so the number of long edges is $u(f) \geq \frac{\sqrt{48n}}{12} |f|$. From Observation 8, it follows that G' has at most $2|f| - 3$ edges. As a consequence, there is a pair of vertices on the boundary of f such that the number of long edges in its bundle is at least

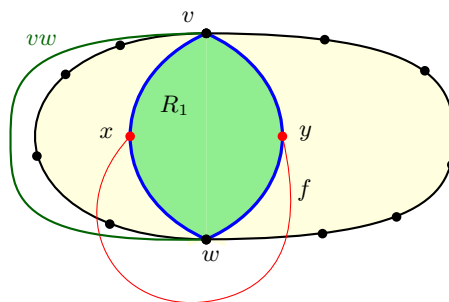
$$\frac{1}{(2|f| - 3)} \frac{\sqrt{48n}}{12} |f| > \frac{\sqrt{48n}}{24}.$$

This implies that there are two vertices, say v and w , to which more than $\frac{\sqrt{48n}}{24}$ vertices inside f have two plane incident edges. We call the set of vertices in $U(f)$ that have plane edges to both vertices v and w the set U_{vw} . This set is marked in Figure 4 (left). We denote the subdrawing of D induced by U_{vw} by D_{vw} ; see Figure 4 (right).



■ **Figure 4** The subdrawing D' induced by U_{vw} and the edges in D_{vw} . Left: The set U_{vw} . Right: The edges adjacent to the leftmost vertex, v_1 , are drawn (in red).

We show that all edges between vertices in U_{vw} cross the edge vw . Let x and y be two vertices of D_{vw} . Let R_1 be the region bounded by the edges xv , vy , yw , and wx that lies inside the face f ; see Figure 5. We show that xy and vw lie completely outside R_1 . The edge xy has to lie either completely inside or completely outside R_1 , because it is adjacent to all edges on the boundary of R_1 . As M is maximal and the edge xy connects two unmatched vertices, it has to cross at least one matching edge. Thus, xy has to lie completely outside R_1 . (There can be no matching edges in R_1 , as R_1 is contained inside the face f .) As H is a maximal plane subdrawing, vw cannot lie inside the face f and thus has to be outside R_1 . Since both edges vw and xy lie completely outside R_1 and the vertices along the boundary of R_1 are sorted $vxwy$, the two edges have to cross. Thus, all edges of D_{vw} cross the edge vw .



■ **Figure 5** The edge xy has to cross the edge vw .

Since the edges from vertices in U_{vw} to v and w are plane, it follows from Lemma 4 that D_{vw} is weakly isomorphic to a generalized twisted drawing. Thus, D_{vw} contains at least $\lfloor \frac{1}{2} \frac{\sqrt{48n}}{24} \rfloor$ pairwise disjoint edges by Theorem 3. Hence, D contains at least $\lfloor \sqrt{\frac{n}{48}} \rfloor$ pairwise disjoint edges. ◀

4 Plane paths in simple drawings

In the previous section, we used generalized twisted drawings to improve the lower bound on the number of disjoint edges in simple drawings of K_n . In this section, we show that generalized twisted drawings are also helpful to improve the lower bound on the length of the longest path in such drawings, where the length of a path is the number of its edges, to $\Omega(\frac{\log n}{\log \log n})$. This improves the previously known best bound of $\Omega((\log n)^{\frac{1}{6}})$, which follows from a result of Pach, Solymosi, and Tóth [15].

► **Theorem 10.** *Every simple drawing D of K_n contains a plane path of length $\Omega(\frac{\log n}{\log \log n})$.*

To prove the new lower bound, we first show that all c -monotone drawings on n vertices contain either a generalized twisted drawing on \sqrt{n} vertices or a drawing weakly isomorphic to an x -monotone drawing on \sqrt{n} vertices. We know that drawings weakly isomorphic to generalized twisted drawings or x -monotone drawings contain plane Hamiltonian paths (by Theorem 3 and Observation 11 below). We conclude that c -monotone drawings contain plane paths of the desired size. We then show that every simple drawing of the complete graph contains either a c -monotone drawing or a plane d -ary tree. With easy observations about the length of the longest path in d -ary trees and by putting all results together, we obtain that every simple drawing D of K_n contains a plane path of length $\Omega(\frac{\log n}{\log \log n})$.

4.1 Plane paths in c -monotone drawings

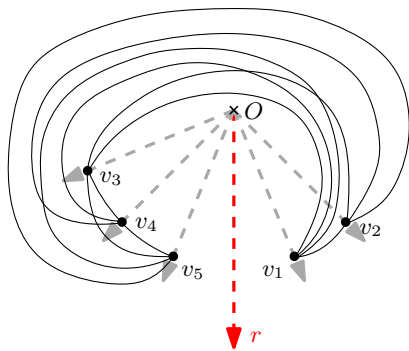
A simple drawing is x -monotone if any vertical line intersects any edge of the drawing at most once (see Figure 6b). This family of drawings has been studied extensively in the literature (see for example [2, 5, 7, 11, 17]). By definition, c -monotone drawings in which there exists a ray emanating from O , which crosses all edges of the drawing, are generalized twisted. In contrast, consider a c -monotone drawing D such that there exists a ray r emanating from O that crosses no edge of D . Then it is easy to see that D is strongly isomorphic to an x -monotone drawing. (A c -monotone drawing on the sphere can be cut along the ray r and the result drawn on the plane such that all rays are vertical lines and the ray r is to the very left of the drawing.) Figure 6a shows a c -monotone drawing D of K_5 where no edge crosses the ray r , and Figure 6b shows an x -monotone drawing of K_5 strongly isomorphic to D . We will call simple drawings that are strongly isomorphic to x -monotone drawings *monotone* drawings. In particular, any c -monotone drawing for which there exists a ray emanating from O that crosses no edge of the drawing is monotone.

It is well-known that any x -monotone drawing of K_n contains a plane Hamiltonian path. For instance, assuming that the vertices are ordered by increasing x -coordinates, the set of edges $v_1v_2, v_2v_3 \dots, v_{n-1}v_n$ form a plane Hamiltonian path.

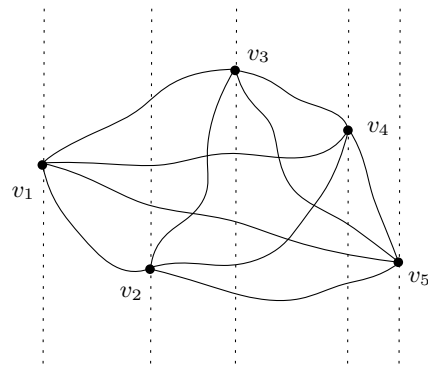
► **Observation 11.** *Every monotone drawing of K_n contains a plane Hamiltonian path.*

We will show that c -monotone drawings contain plane paths of size \sqrt{n} , by showing that any c -monotone drawing of K_n contains a subdrawing of $K_{\sqrt{n}}$ that is either generalized twisted or monotone. To do so, we will use Dilworth's Theorem on chains and anti-chains in partially ordered sets. A *chain* is a subset of a partially ordered set such that any two distinct elements are comparable. An *anti-chain* is a subset of a partially ordered set such that any two distinct elements are incomparable.

► **Theorem 12** (Dilworth's Theorem, [8]). *Let P be a partially ordered set of at least $(s-1)(t-1)+1$ elements. Then P contains a chain of size s or an antichain of size t .*

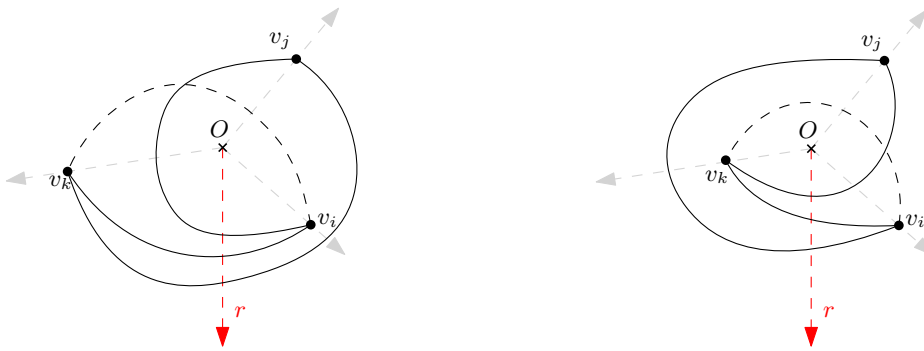


(a) A c -monotone drawing D of K_5 such that the ray r crosses no edge of D .



(b) An x -monotone drawing of K_5 strongly isomorphic to D of Figure 6a.

■ **Figure 6** Two strongly isomorphic monotone drawings of K_5 .



■ **Figure 7** If edges $v_i v_j$ and $v_j v_k$ cross r in a c -monotone drawing, then $v_i v_k$ must also cross r .

► **Theorem 13.** *Let s, t be two integers, $1 \leq s, t \leq n$, such that $(s - 1)(t - 1) + 1 \leq n$. Let D be a c -monotone drawing of K_n . Then D contains either a generalized twisted drawing of K_s or a monotone drawing of K_t as subdrawing. In particular, if $s = t = \lceil \sqrt{n} \rceil$, D contains a complete subgraph K_s whose induced drawing is either generalized twisted or monotone.*

Proof Sketch. Without loss of generality we may assume that the vertices of D appear counterclockwise around O in the order v_1, v_2, \dots, v_n . Let r be a ray emanating from O , keeping v_1 and v_n on different sides. We define an order, \preceq , in this set of vertices as follows: $v_i \preceq v_j$ if and only if either $i = j$ or $i < j$ and the edge (v_i, v_j) crosses r .

We show that \preceq is a partial order. The relation is clearly reflexive and antisymmetric. Besides, if $v_i \preceq v_j$ and $v_j \preceq v_k$, then $v_i \preceq v_k$, because $i < j$ and $j < k$ imply $i < k$, and if $v_i v_j$ and $v_j v_k$ cross r , then $v_i v_k$ also crosses r (see Figure 7). Hence, the relation is transitive.

In this partial order \preceq , a chain consists of a subset $v_{i_1}, \dots, v_{i_{s-1}}$ of pairwise comparable vertices, that is, a subset of vertices such that their induced subdrawing is generalized twisted (all edges cross r). An antichain, $v_{j_1}, \dots, v_{j_{t-1}}$, consists of a subset of pairwise incomparable vertices, that is, a subset of vertices such that their induced subdrawing is monotone (no edge crosses r). Therefore, the first part of the theorem follows from applying Theorem 12 to the set of vertices of D and the partial order \preceq .

Finally, observe that if $s = t \leq \lceil \sqrt{n} \rceil$, then $(s - 1)(t - 1) + 1 \leq n$. Thus, D contains a complete subgraph $K_{\lceil \sqrt{n} \rceil}$ whose induced subdrawing is either generalized twisted or monotone. ◀

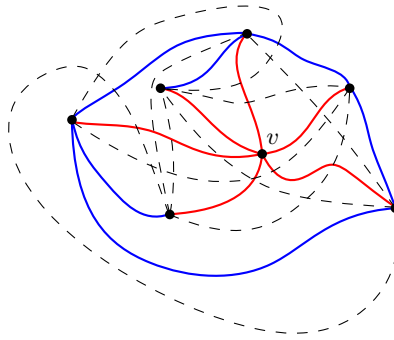
Combining Theorems 3 and 13 with Observation 11, we obtain the following theorem.

► **Theorem 14.** *Every c -monotone drawing of K_n contains a plane path of length $\Omega(\sqrt{n})$.*

4.2 Plane paths in simple drawings

To show that any simple drawing of K_n contains a plane path of length $\Omega(\frac{\log n}{\log \log n})$, we will use d -ary trees. A d -ary tree is a rooted tree in which no vertex has more than d children. It is well-known that the height of a d -ary tree on n vertices is $\Omega(\frac{\log n}{\log d})$.

Proof of Theorem 10. Let v be a vertex of D and let $S(v)$ be the star centered at v , that is, the set of edges of D incident to v . $S(v)$ can be extended to a maximal plane subdrawing H that must be biconnected by Theorem 5. See Figure 8 for a depiction of $S(v)$ and H .



■ **Figure 8** A simple drawing of K_7 . The red edges show the star $S(v)$, the red and blue edges together form a maximal plane subdrawing H . Dashed edges are edges of K_7 that are not in H .

Assume first that there is a vertex w in $H \setminus v$ that has degree at least $(\log n)^2$ in H . Let U_{vw} be the set of vertices neighbored in H to both, v and w . Note that $|U_{vw}| \geq (\log n)^2$. The subdrawing H' of H consisting of the vertices in U_{vw} , the vertices v , and w , and the edges from v to vertices in U_{vw} , and from w to vertices in U_{vw} is a plane drawing of $K_{2,|U_{vw}|}$. From Lemma 4, it follows that the subdrawing of D induced by U_{vw} is weakly isomorphic to a c -monotone drawing. Therefore, by Theorem 14, the subdrawing induced by U_{vw} contains a plane path of length $\Omega(\sqrt{|U_{vw}|}) = \Omega(\log n)$.

Assume now that the maximum degree in $H \setminus v$ is less than $(\log n)^2$. Since H is biconnected, $H \setminus v$ contains a plane tree T of order $n - 1$ whose maximum degree is at most $(\log n)^2$. Thus, considering that T is rooted, the diameter of T is at least $\Omega(\frac{\log n}{\log \log n})$. Therefore, since T is plane, it contains a plane path of length at least $\Omega(\frac{\log n}{\log \log n})$ and the theorem follows. ◀

5 Characterizing generalized twisted drawings

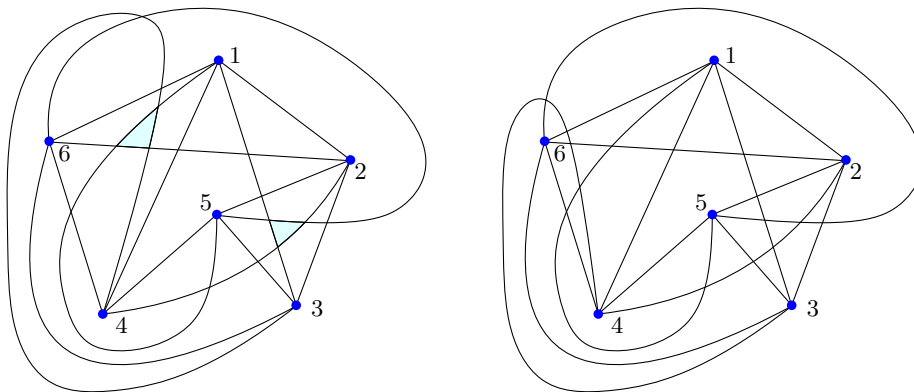
In previous sections, we have seen how generalized twisted drawings were used to make progress on open problems of simple drawings. In addition to this, generalized twisted drawings are also interesting in their own right and have some quite surprising structural properties. Despite the fact that research on generalized twisted drawings is rather recent and still ongoing, there are already several interesting characteristics and structural results. Some of them will be presented in this section.

One characterization involves curves crossing every edge once. From the definition of generalized twisted drawing (see Figure 1), there always exists a simple curve that crosses all edges of the drawing exactly once (for instance, a curve that starts at O and follows r until

it reaches a point Z on r in the unbounded cell). In Theorem 15, we show that the converse is also true. That is, every simple drawing D of K_n in which we can add a simple curve that crosses every edge of D exactly once is weakly isomorphic to a generalized twisted drawing.

Another characterization is based on what we call *antipodal vi-cells*. For any three vertices in a simple drawing D of K_n , the three edges connecting them form a simple cycle which we call a *triangle*. Every such triangle partitions the plane (or sphere) into two disjoint regions which are the *sides* of the triangle (in the plane a bounded and an unbounded one). Two cells of D are called *antipodal* if for each triangle of D , they lie on different sides. Further, we call a cell with a vertex on its boundary a vertex-incident-cell or, for short, a *vi-cell*.

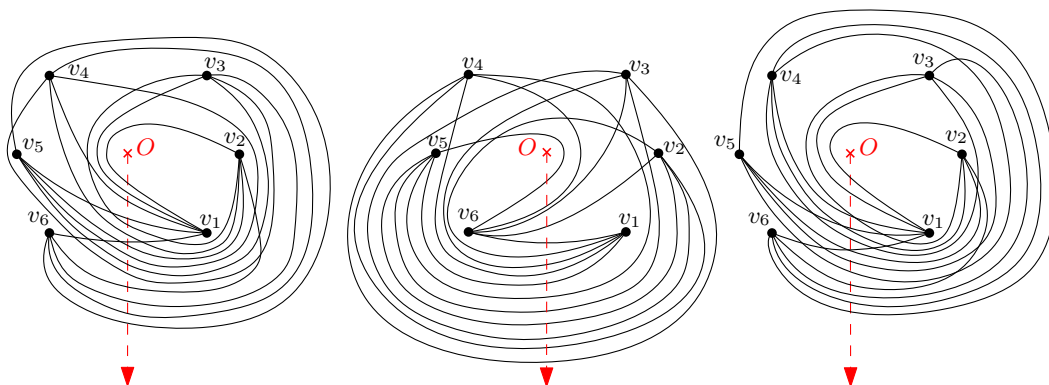
By definition, every generalized twisted drawing D contains two antipodal cells, namely, the cell containing the starting point of the ray r and the unbounded cell. This follows from the fact that the ray r crosses every edge exactly once. Hence, r crosses the boundary of any triangle exactly three times, so the cells containing the “endpoints” of r must be on different sides of the triangle.



■ **Figure 9** Two weakly isomorphic drawings of K_6 that are not weakly isomorphic to any generalized twisted drawing. Antipodal cells are marked in blue.

It turns out that the converse (existence of two antipodal cells implies weakly isomorphic to generalized twisted) is not true. Figure 9 (left) shows a drawing of K_6 that contains two antipodal cells, but no antipodal vi-cells. From Theorem 15 below it will follow that such drawings cannot be weakly isomorphic to a generalized twisted drawing. However, we observed that for all generalized twisted drawings of K_n with $n \leq 6$, both, the cell containing the startpoint of the ray r and the unbounded cell, are vi-cells. Figure 10 shows all (up to strong isomorphism) simple drawings of K_6 that are weakly isomorphic to generalized twisted drawings. We show that this is true in general. More than that, we show in Theorem 16 that every drawing of K_n that is weakly isomorphic to a generalized twisted drawing contains a pair of antipodal vi-cells. In the other direction, we show in Theorem 15 that every simple drawing containing a pair of antipodal vi-cells is weakly isomorphic to a generalized twisted drawing.

The final characterization is based on the extension of a given drawing of the complete graph to a drawing containing a spanning, plane bipartite graph that has all vertices of the original drawing on one side of the bipartition. From the definition of generalized twisted drawings, it follows that any generalized twisted drawing D of K_n can be extended to a simple drawing D' of K_{n+2} including new vertices O and Z such that D' contains a plane drawing of a spanning bipartite graph. One side of the bipartition consists of all vertices in D and the other side of the bipartition consists of the new vertices O and Z . Moreover,



■ **Figure 10** All different generalized twisted drawings of K_6 (up to weak isomorphism). The rightmost drawing is twisted.

the edge OZ crosses all edges of D . One way to add the new vertices and edges incident to them is to draw (1) the vertex O at point O , (2) the vertex Z in the unbounded cell on the ray r , (3) the edge OZ straight-line (along the ray r), (4) edges from O to the vertices of D straight-line (along the inner segment of the rays crossing through the vertices), and (5) edges from Z to the vertices of D first far away in a curve and the final part straight-line (along the outer segment of the rays crossing through the vertices). The converse, that every drawing that can be extended like this is weakly isomorphic to a generalized twisted drawing, has already been shown in Lemma 4.

We show the following characterizations.

► **Theorem 15** (Characterizations of generalized twisted drawings). *Let D be a simple drawing of K_n . Then, the following properties are equivalent.*

Property 1 D is weakly isomorphic to a generalized twisted drawing.

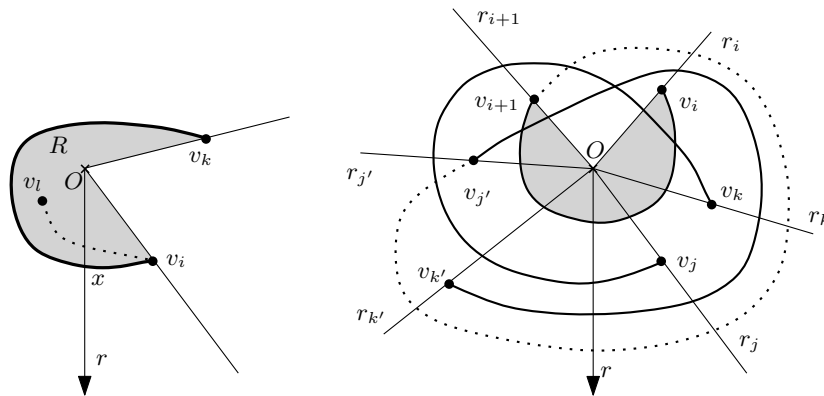
Property 2 D contains two antipodal vi-cells.

Property 3 D can be extended by a simple curve c such that c crosses every edge of D exactly once.

Property 4 D can be extended by two vertices, O and Z , and edges incident to the new vertices such that D together with the new vertices and edges is a simple drawing of K_{n+2} , the edge OZ crosses every edge of D , and no edge incident to O crosses any edge incident to Z .

To prove Theorem 15, we will first show that Property 1 implies Property 2 (Theorem 16). We next show that Property 2 implies Property 3 (Theorem 17). Then, we show that Property 3 implies Property 4 (Theorem 18). By Lemma 4, Property 4 implies Property 1. Thus, all properties are equivalent. In a full version of this work, we will extend the theorem to show that also strong isomorphism to a generalized twisted drawing is equivalent to the properties of Theorem 15. We show this by proving that any simple drawing of K_n fulfilling Property 4 is strongly isomorphic to a generalized twisted drawing. However, the reasoning for strong isomorphism is quite lengthy and would exceed the space constraints of this submission.

► **Theorem 16.** *Every simple drawing of K_n which is weakly isomorphic to a generalized twisted drawing of K_n , with $n \geq 3$, contains a pair of antipodal vi-cells. In generalized twisted drawings the cell containing O and the unbounded cell form such a pair.*



■ **Figure 11** Left: If there is a vertex v_l in R , it cannot be connected to v_i without crossing r before x . Right: If the edge $v_j v_k$ crosses the segment $\overline{Ov_i}$ and the edge $v_{j'} v_{k'}$ crosses the segment $\overline{Ov_{i+1}}$, then there is no way of connecting v_{i+1} and $v_{j'}$.

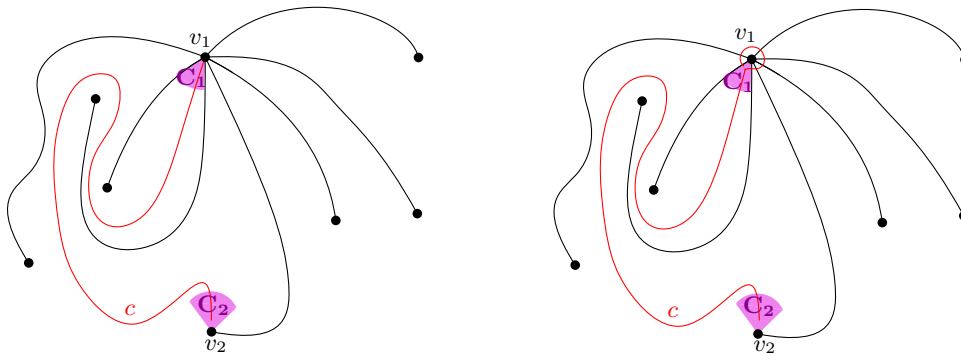
Proof sketch. We first show that every generalized twisted drawing D of K_n , with $n \geq 3$, contains a pair of antipodal vi-cells, where O lies in a cell of that pair. Let c be the segment OZ , where Z is a point on r in the unbounded cell. By definition of generalized twisted, c crosses every edge of D once, so O and Z are in two antipodal cells C_1 and C_2 , respectively.

To prove that C_1 is a vi-cell, we use the following properties. First, if we take the first edge $v_i v_k$ that crosses c (as seen from O) at point x , then we can prove that $k = i + 1$ and the bounded region R defined by the edge $v_i v_{i+1}$ and the segments $\overline{Ov_i}$ and $\overline{Ov_{i+1}}$ is empty (see Figure 11, left). Second, using this empty region we can prove that D cannot contain simultaneously an edge $v_j v_k$ crossing $\overline{Ov_i}$ and another edge $v_{j'} v_{k'}$ crossing $\overline{Ov_{i+1}}$ (see Figure 11, right). Therefore, at least one of the segments $\overline{Ov_i}$ and $\overline{Ov_{i+1}}$ is uncrossed, and O necessarily lies in a vi-cell (with either v_i or v_{i+1} on the boundary). Finally, arguing on the last edge crossing c and the unbounded cell, we can show that Z also lies in a vi-cell.

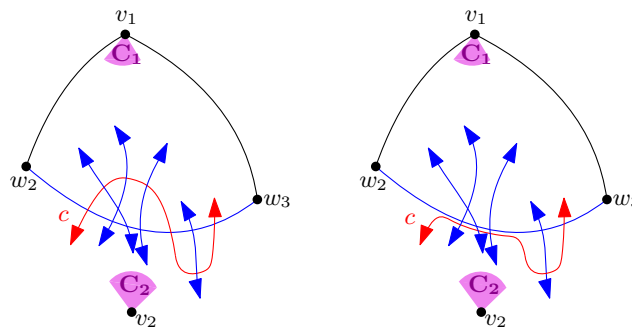
To show that also every drawing which is weakly isomorphic to a generalized twisted drawing contains a pair of antipodal vi-cells, we use Gioan’s Theorem [6, 14]. By Gioan’s Theorem, any two weakly isomorphic drawings of K_n can be transformed into each other with a sequence of triangle-flips and at most one reflection of the drawing. A *triangle-flip* is an operation which transforms a triangular cell Δ that has no vertex on its boundary by moving one of its edges across the intersection of the two other edges of Δ . We show that if a drawing D_1 contains two antipodal vi-cells, then after performing a triangle flip on D_1 , the resulting drawing D_2 still has two antipodal vi-cells. The main argument is that triangle-flips are only applied to cells without vertices on their boundary, and thus the antipodality of the vi-cells cannot change. ◀

► **Theorem 17.** *In any simple drawing D of K_n that contains a pair of antipodal vi-cells, it is possible to draw a curve c that crosses every edge of D exactly once.*

Proof sketch. Let (C_1, C_2) be a pair of antipodal vi-cells of D . Let v_1 be a vertex on the boundary of C_1 and v_2 a vertex on the boundary of C_2 . We construct the curve as follows: First, we draw a simple curve c from C_1 to C_2 such that (1) it emanates from v_1 in C_1 and ends in C_2 very close to v_2 , (2) does not cross any edge incident to v_1 , (3) only intersects edges of D in proper crossings, and (4) has the minimum number of crossings with edges of D among all curves that fulfill (1), (2) and (3). This curve c always exists since $S(v_1)$ is a plane drawing that has only a face in which both v_1 and v_2 lie (see Figure 12, left).



■ **Figure 12** Building a curve such that it crosses every edge of D once and its endpoints do not lie on any edges or vertices of D .



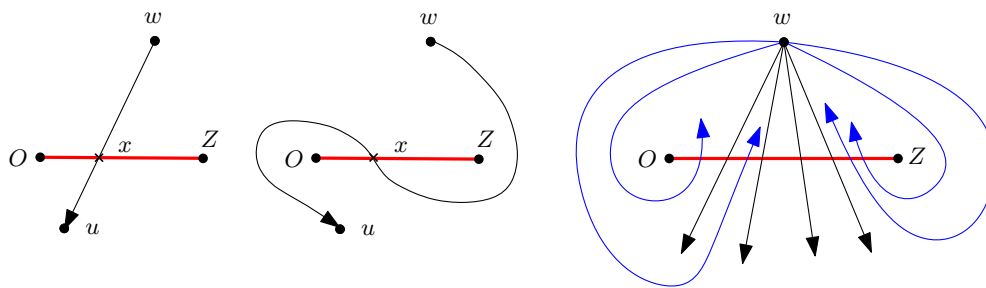
■ **Figure 13** Decreasing the number of crossings between c and the edge w_2w_3 .

Then, we prove that c crosses every edge w_2w_3 in D that is not incident to v_1 exactly once. On the one hand, since c connects two antipodal cells, the endpoints of c have to be on two different sides of the triangle T formed by v_1 , w_2 and w_3 . Thus, c has to cross w_2w_3 an odd number of times because it does not cross $S(v_1)$ and must cross the boundary of T an odd number of times. On the other hand, if c crosses w_2w_3 at least three times, then we can prove that c can be redrawn as shown in Figure 13, decreasing the number of crossings, which contradicts (4). Therefore, c crosses every edge w_2w_3 at most twice and, consequently, only once.

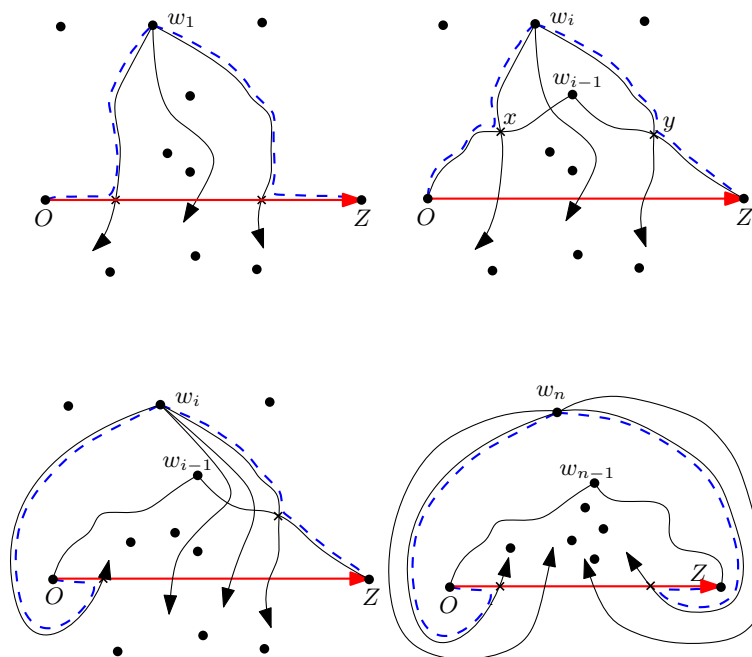
Finally, we change the end of c from v_1 to a point in C_1 in the following way (see Figure 12, right). From some point of c sufficiently close to v_1 and inside C_1 , we reroute c by going around v_1 such that only the edges incident to v_1 are crossed, and end at a point in C_1 . ◀

► **Theorem 18.** *Let D be a simple drawing of K_n in which it is possible to draw a simple curve c that crosses every edge of D exactly once. Then, D can be extended by two vertices O and Z (at the position of the endpoints of the curve), and edges incident to those vertices such that the obtained drawing is a simple drawing of K_{n+2} , no edge incident to O crosses any edge incident to Z , and all edges in D cross the edge OZ .*

Proof sketch. Let $c = OZ$ be the curve crossing every edge of D once, oriented from O to Z . Let wu be an edge of D , oriented from w to u , crossing OZ at a point x . We say that wu is a *top* (respectively *bottom*) edge if the clockwise order of w, Z, u and O around x is w, Z, u, O (respectively w, O, u, Z). See Figure 14. With these definitions, we can prove that there is a vertex w_1 in D such that all the oriented edges emanating from w_1 are top in relation to c .



■ **Figure 14** Top and bottom edges. For simplicity, the curve OZ is drawn as a horizontal line. Left: A top edge wu . Centre: A bottom edge wu . Right: The (black) top and (blue) bottom edges of $S(w)$.



■ **Figure 15** Building the (dashed) edges w_iO and w_iZ .

Thus, by removing w_1 and all its incident edges from D , there is a vertex w_2 in the new drawing such that all its incident edges are top, and so on. As a consequence, there is a natural order w_1, w_2, \dots, w_n of the vertices of D such that for any vertex w_i , the edges w_iw_j with $j > i$ are top, and the edges w_iw_j with $j < i$ are bottom.

Given the natural order w_1, w_2, \dots, w_n , our construction of the extended drawing is as follows. Let D'_0 be the simple drawing formed by the vertices and edges of D , O and Z as new vertices, and c as the edge connecting O and Z . From D'_0 , we build new drawings D'_1, D'_2, \dots, D'_n , by adding in step i the edges w_iO and w_iZ . These two edges are added very close to some edges in D'_{i-1} . Figure 15 illustrates how these two edges are added in each step.

In the first step, the edge Ow_1 follows the curve OZ until the crossing point between OZ and the first top edge w_1u emanating from w_1 , and then it follows this top edge until reaching w_1 . The edge Zw_1 is built in an analogous way, taking the last top edge emanating from w_1 . See Figure 15 top-left. For $i = 2, \dots, n-1$, in step i we do different constructions depending on whether the first and last top edges of $S(w_i)$ cross the edges $w_{i-1}O$ and $w_{i-1}Z$. If the first top edge w_iu_1 crosses $w_{i-1}O$ at a point x and the last top edge w_iu_k crosses $w_{i-1}Z$ at a point y (see Figure 15 top-right), then Ow_i follows Ow_{i-1} until x , and then it follows u_1w_i until w_i . The edge Zw_i is built following Zw_{i-1} until y and then following u_kw_i . On the contrary, if the first and the last top edges of $S(w_i)$ only cross one of $w_{i-1}O$ and $w_{i-1}Z$, say $w_{i-1}Z$ (see Figure 15 bottom-left), then Ow_i follows OZ until the crossing point between OZ and the last bottom edge of $S(w_i)$, and then it follows this bottom edge until w_i . The edge Zw_i is built as in the first step, using the last top edge of $S(w_i)$. In the last step, we build Ow_n and Zw_n as in the first step, but using the first and the last bottom edges of $S(w_n)$ instead of the first and last top edges. See Figure 15 bottom-right.

By a detailed analysis of cases, we can prove for $i = 1, \dots, n$ that D'_i is a simple drawing such that no edge incident to O crosses any edge incident to Z . Therefore, D'_n is the drawing of K_{n+2} satisfying the required properties. ◀

6 Conclusion and outlook

Generalized twisted drawings have a surprisingly rich structure and many useful properties. We showed several of those properties in Section 2 and different characterizations of generalized twisted drawings in Section 5. We have proven in Section 2 that every generalized twisted drawing on an odd number of vertices contains a plane Hamiltonian cycle, and therefore one especially interesting open question is the following.

► **Conjecture 19.** *Every generalized twisted drawing of K_n contains a plane Hamiltonian cycle.*

Using properties of generalized twisted drawings has turned out to be helpful for investigating simple drawings in general. We first improved the lower bound on the number of disjoint edges in simple drawings of K_n to $\Omega(\sqrt{n})$ (Section 3). Then generalized twisted drawings played the central role to improve the lower bound on the length of plane paths contained in every simple drawing of K_n to $\Omega(\frac{\log n}{\log \log n})$ (Section 4).

On the other hand, from Theorem 17 it immediately follows that no drawing that is weakly isomorphic to a generalized twisted drawing can contain three interior-disjoint triangles (since the endpoints of the curve crossing every edge once must be on opposite sides of every triangle, the maximum number of interior-disjoint triangles is two). Up to strong isomorphism, there are only two simple drawings of K_4 . The plane drawing contains three interior-disjoint triangles. Thus, (up to strong isomorphism) the only drawing of K_4 that is weakly isomorphic to a generalized twisted drawing, is the drawing with a crossing. Hence, in every generalized twisted drawing all subdrawings induced by 4 vertices contain a crossing and thus every generalized twisted drawing is crossing maximal. Up to strong isomorphism, there are two crossing maximal drawings of K_5 : the convex drawing of K_5 and the twisted drawing of K_5 . Since the convex drawing contains three interior-disjoint triangles, the only (up to strong isomorphism) drawing of K_5 that is weakly isomorphic to a generalized twisted drawing is the twisted drawing of K_5 (that is drawn generalized twisted in Figure 1).

It is part of our ongoing work to show that for $n \geq 7$, a drawing is weakly isomorphic to a generalized twisted drawing if and only if all subdrawings induced by five vertices are weakly isomorphic to the twisted K_5 . Interestingly, the $n \geq 7$ is necessary as there is a drawing

with 6 vertices that contains only twisted drawings of K_5 but is not weakly isomorphic to a generalized twisted drawing (see the drawings in Figure 9). There are (up to strong isomorphism) three more simple drawings of K_6 that consist of only twisted drawings of K_5 and they are all weakly isomorphic to generalized twisted drawings (see Figure 10).

References

- 1 Bernardo M. Ábrego, Oswin Aichholzer, Silvia Fernández-Merchant, Thomas Hackl, Jürgen Pammer, Alexander Pilz, Pedro Ramos, Gelasio Salazar, and Birgit Vogtenhuber. All good drawings of small complete graphs. In *Proc. 31st European Workshop on Computational Geometry EuroCG '15*, pages 57–60, Ljubljana, Slovenia, 2015. URL: <http://www.ist.tu-graz.ac.at/files/publications/geometry/aafhprsv-agdsc-15.pdf>.
- 2 Bernardo M. Ábrego, Oswin Aichholzer, Silvia Fernández-Merchant, Pedro Ramos, and Gelasio Salazar. Shellable drawings and the cylindrical crossing number of K_n . *Discrete & Computational Geometry*, 52(4):743–753, 2014. doi:10.1007/s00454-014-9635-0.
- 3 Oswin Aichholzer, Alfredo García, Javier Tejel, Birgit Vogtenhuber, and Alexandra Weinberger. Plane matchings in simple drawings of complete graphs. In *Abstracts of the Computational Geometry: Young Researchers Forum*, pages 6–10, 2021. URL: <https://cse.buffalo.edu/socg21/files/YRF-Booklet.pdf#page=6>.
- 4 Oswin Aichholzer, Alfredo García, Javier Tejel, Birgit Vogtenhuber, and Alexandra Weinberger. Plane paths in simple drawings of complete graphs. In *Abstracts of XIX Encuentros de Geometría Computacional*, page 4, 2021. URL: https://quantum-explore.com/wp-content/uploads/2021/06/Actas_egc21.pdf#page=11.
- 5 Oswin Aichholzer, Thomas Hackl, Alexander Pilz, Gelasio Salazar, and Birgit Vogtenhuber. Deciding monotonicity of good drawings of the complete graph. In *Abstracts XVI Spanish Meeting on Computational Geometry (XVI EGC)*, pages 33–36, 2015.
- 6 Alan Arroyo, Dan McQuillan, R. Bruce Ritcher, and Gelasio Salazar. Drawings of K_n with the same rotation scheme are the same up to Reidemeister moves (Gioan’s theorem). *Australasian Journal of Combinatorics*, 67:131–144, 2017.
- 7 Martin Balko, Radoslav Fulek, and Jan Kynčl. Crossing numbers and combinatorial characterization of monotone drawings of K_n . *Discrete Comput. Geom.*, 53(1):107–143, 2015. doi:10.1007/s00454-014-9644-z.
- 8 Robert P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51(1):161–166, 1950. doi:10.2307/1969503.
- 9 Jacob Fox and Benny Sudakov. Density theorems for bipartite graphs and related ramsey-type results. *Combinatorica*, 29(2):153–196, 2009. doi:10.1007/s00493-009-2475-5.
- 10 Radoslav Fulek. Estimating the number of disjoint edges in simple topological graphs via cylindrical drawings. *SIAM Journal on Discrete Mathematics*, 28(1):116–121, 2014. doi:10.1137/130925554.
- 11 Radoslav Fulek, Michael J. Pelsmajer, Marcus Schaefer, and Daniel Štefankovič. Hanani-Tutte, monotone drawings, and level-planarity. In *Thirty essays on geometric graph theory*, pages 263–287. Springer, New York, NY, 2013. doi:10.1007/978-1-4614-0110-0_14.
- 12 Radoslav Fulek and Andres J. Ruiz-Vargas. Topological graphs: empty triangles and disjoint matchings. In *Proceedings of the 29th Annual Symposium on Computational Geometry (SoCG'13)*, pages 259–266, 2013. doi:10.1145/2462356.2462394.
- 13 Alfredo García, Alexander Pilz, and Javier Tejel. On plane subgraphs of complete topological drawings. *ARS MATHEMATICA CONTEMPORANEA*, 20:69–87, 2021. doi:10.26493/1855-3974.2226.e93.
- 14 Emeric Gioan. Complete graph drawings up to triangle mutations. In *Graph-Theoretic Concepts in Computer Science. WG 2005. Lecture Notes in Computer Science, vol 3787*, pages 139–150. Springer, 2005. doi:10.1007/11604686_13.

- 15 János Pach, József Solymosi, and Géza Tóth. Unavoidable configurations in complete topological graphs. *Discrete Comput Geometry*, 30:311–320, 2003. doi:10.1007/s00454-003-0012-9.
- 16 János Pach and Géza Tóth. Disjoint edges in topological graphs. In *Proceedings of the 2003 Indonesia-Japan Joint Conference on Combinatorial Geometry and Graph Theory (IJC-CGGT'03)*, volume 3330, pages 133–140, 2005. doi:10.1007/978-3-540-30540-8_15.
- 17 János Pach and Géza Tóth. Monotone crossing number. In *Graph Drawing*, pages 278–289. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-25878-7_27.
- 18 Nabil H. Rafla. *The good drawings D_n of the complete graph K_n* . PhD thesis, McGill University, Montreal, 1988. URL: https://escholarship.mcgill.ca/concern/file_sets/cv43nx65m?locale=en.
- 19 Andres J. Ruiz-Vargas. Empty triangles in complete topological graphs. In *Discrete Computational Geometry*, volume 53, pages 703–712, 2015. doi:10.1007/s00454-015-9671-4.
- 20 Andres J. Ruiz-Vargas. Many disjoint edges in topological graphs. *Computational Geometry*, 62:1–13, 2017. doi:10.1016/j.comgeo.2016.11.003.
- 21 Andrew Suk. Disjoint edges in complete topological graphs. *Discrete & Computational Geometry*, 49(2):280–286, 2013. doi:10.1007/s00454-012-9481-x.

Edge Partitions of Complete Geometric Graphs

Oswin Aichholzer  

Institute of Software Technology,
Technische Universität Graz, Austria

Johannes Obenaus  

Department of Computer Science,
Freie Universität Berlin, Germany

Joachim Orthaber  

Institute of Software Technology,
Technische Universität Graz, Austria

Rosna Paul  

Institute of Software Technology,
Technische Universität Graz, Austria

Patrick Schnider   

Department of Mathematical Sciences,
University of Copenhagen, Denmark

Raphael Steiner   

Department of Computer Science,
ETH Zürich, Switzerland

Tim Taubner  

Department of Computer Science,
ETH Zürich, Switzerland

Birgit Vogtenhuber  

Institute of Software Technology,
Technische Universität Graz, Austria

Abstract

In this paper, we disprove the long-standing conjecture that any complete geometric graph on $2n$ vertices can be partitioned into n plane spanning trees. Our construction is based on so-called bumpy wheel sets. We fully characterize which bumpy wheels can and in particular which *cannot* be partitioned into plane spanning trees (or even into arbitrary plane *subgraphs*).

Furthermore, we show a sufficient condition for *generalized wheels* to not admit a partition into plane spanning trees, and give a complete characterization when they admit a partition into plane spanning double stars.

Finally, we initiate the study of partitions into beyond planar subgraphs, namely into k -planar and k -quasi-planar subgraphs and obtain first bounds on the number of subgraphs required in this setting.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory

Keywords and phrases edge partition, complete geometric graph, plane spanning tree, wheel set

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.6

Related Version *Full Version:* <https://arxiv.org/abs/2108.05159>

Full Version: <https://arxiv.org/abs/2112.08456>

Funding *Oswin Aichholzer:* Partially supported by the Austrian Science Fund (FWF): W1230 and the European Union H2020-MSCA-RISE project 73499 – CONNECT.

Johannes Obenaus: Supported by ERC StG 757609.

Rosna Paul: Supported by the Austrian Science Fund (FWF): W1230.

Patrick Schnider: Supported by ERC StG 716424 – CAsE.

Raphael Steiner: Supported by an ETH Zurich Postdoctoral Fellowship.

Birgit Vogtenhuber: Partially supported by the Austrian Science Fund (FWF): I 3340-N35.

Acknowledgements Research on this work has been initiated in March 2021, at the 5th research workshop of the collaborative D-A-CH project *Arrangements and Drawings*, which was funded by the DFG, the FWF, and the SNF. We thank the organizers and all participants for fruitful discussions. Further, we thank the anonymous reviewers for their insightful comments and suggestions.



© Oswin Aichholzer, Johannes Obenaus, Joachim Orthaber, Rosna Paul, Patrick Schnider, Raphael Steiner, Tim Taubner, and Birgit Vogtenhuber; licensed under Creative Commons License CC-BY 4.0

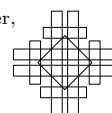
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 6; pp. 6:1–6:16



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

A geometric graph $G = G(P, E)$ is a drawing of a graph in the plane where the vertex set is drawn as a point set P in general position (that is, no three points are collinear) and each edge of E is drawn as a straight-line segment between its vertices. A geometric graph G is *plane* if no two of its edges *cross* (that is, share a point in their relative interior). A *partition* (also called edge partition) of a graph G is a set of edge-disjoint subgraphs of G whose union is G . A subgraph of (a connected graph) G is *spanning* if it is connected and its vertex set is the same as the one of G . In 2003, Ferran Hurtado shared the following long-standing open question, which has commonly been conjectured to have a positive answer (see [9, 6]):

► **Question 1** ([6]). *Can every complete geometric graph on $2n$ vertices be partitioned into n plane spanning trees?*

Note that with $2n > 0$ vertices, the complete graph has exactly the right number of edges to admit a partition into n spanning trees, while this is not the case for $2n + 1$ vertices. In the following, we consider complete geometric graphs to have $2n$ vertices unless stated otherwise. Further, we denote the complete geometric graph on a point set P as $K(P)$.

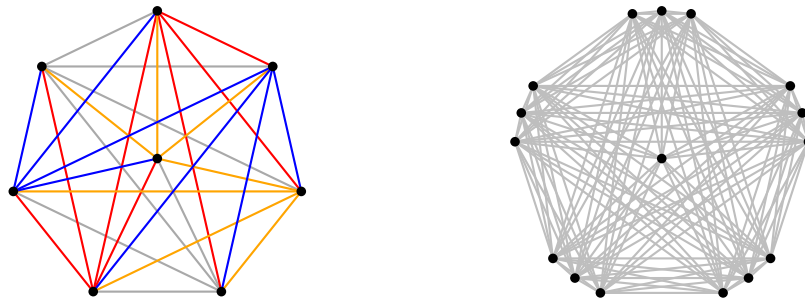
Related work. Several approaches have been made to answer Question 1. When P is in convex position it follows from a result of Bernhart and Kainen [4] that $K(P)$ can be partitioned into plane spanning paths, implying a positive answer. Further, Bose et al. [6] gave a complete characterization of all possible partitions into plane spanning trees for convex point sets. Similarly, when $P = W_{2n}$ is a *regular wheel set* (the vertex set of a regular $(2n - 1)$ -gon plus its center), Aichholzer et al. [2] showed how to partition $K(P)$ into plane spanning *double stars* (trees with at most two vertices of degree ≥ 2), and Trao et al. [14] recently characterized all possible partitions (into arbitrary plane spanning trees). Further, Aichholzer et al. [2] provide a positive answer to Question 1 for all point sets of (even) cardinality at most 10, obtained by exhaustive computations.

Relaxing the requirement that the trees must be spanning, Bose et al. [6] showed that if for a general point set P , there exists an arrangement of k lines in which every cell contains at least one point from P , then the complete geometric graph on P admits a partition into $2n - k$ plane trees, k of which are plane double stars. This result implies that Question 1 has a positive answer if P contains n pairwise crossing segments, which is the case if and only if P has exactly n *halving lines* [10] (a line through two points of P is called *halving line* if it has exactly $n - 1$ points of P on either side and the corresponding edge is called *halving edge*).

For the related *packing* problem where not all edges of the underlying graphs must be covered, Biniáz and García [5] showed that $\lfloor n/3 \rfloor$ plane spanning trees can be packed in any complete geometric graph on n vertices, which is currently the best lower bound. Further, in [1] and [2], packing plane spanning graphs with short edges and spanning paths, respectively, have been considered.

Contribution. In this work, we provide a negative answer to Question 1 (refuting the prevalent conjecture). We even provide a negative answer to the following weaker question:

► **Question 2.** *Can every complete geometric graph on $2n$ vertices be partitioned into n plane subgraphs?*



■ **Figure 1** *Left:* A partition of W_8 into $n = 4$ plane spanning trees. *Right:* The bumpy wheel $BW_{5,3}$.

Note that the problem of partitioning a geometric graph into plane subgraphs is equivalent to a classic edge coloring problem, where each edge should be assigned a color in such a way that no two edges of the same color cross (of course using as few colors as possible). This problem received considerable attention from a variety of perspectives (see for example [11] and references therein) and is also the topic of the CG:SHOP challenge 2022 [7].

The point sets in our construction, so-called *bumpy wheel sets*, have been introduced in [12, 13]. For positive odd¹ integers k and ℓ , the *bumpy wheel* $BW_{k,\ell}$ is derived from the regular wheel W_{k+1} by replacing each of the k hull vertices by a *group* of ℓ vertices as follows. All vertices (except the center) lie on the convex hull and the vertices within each group are ε -close for some (small enough) $\varepsilon > 0$. In particular, the convex hull of any $\frac{k+1}{2}$ consecutive groups does not contain the center vertex (see Figure 1 for an illustration). Slightly abusing notation, $BW_{k,\ell}$ refers to the underlying point set as well as the complete geometric graph interchangeably. Note that for $\ell = 1$ we obtain a regular wheel set and for $k = 1$ a point set in convex position and hence we assume $k, \ell \geq 3$ in the following.

Our motivation to study bumpy wheels stemmed from the fact that Schnider [12] showed that $BW_{3,3}$ cannot be partitioned into plane double stars. In contrast, this is always possible for complete geometric graphs on regular wheel sets [2], as well as complete geometric graphs on point sets admitting n pairwise crossing edges [6] (which also includes convex point sets).

Our first main contribution in this work is to fully characterize for which (odd) parameters k and ℓ , the bumpy wheel $BW_{k,\ell}$ can and in particular *cannot* be partitioned into plane spanning trees or plane *subgraphs* (note that also in the setting of partitioning into plane subgraphs we are only interested in partitions into n subgraphs). Surprisingly, allowing arbitrary subgraphs instead of spanning trees does not help much, as it turns out that $BW_{3,5}$ is the only bumpy wheel that can be partitioned into plane subgraphs but not into plane spanning trees.

► **Theorem 3.** *For odd parameters $k, \ell \geq 3$, the edges of $BW_{k,\ell}$ cannot be partitioned into $n = \frac{k\ell+1}{2}$ plane spanning trees if and only if $\ell > 3$.*

► **Theorem 4.** *For odd parameters $k, \ell \geq 3$, the edges of $BW_{k,\ell}$ cannot be partitioned into $n = \frac{k\ell+1}{2}$ plane subgraphs if and only if $\ell > 5$ or ($\ell = 5$ and $k > 3$).*

¹ We require k and ℓ to be odd for an even number of vertices in total (k has to be odd anyway, since otherwise W_{k+1} would not be in general position).

We further consider the more general case of complete geometric graphs on point sets with exactly one point inside the convex hull. In this generalized setting, we show a sufficient condition for the non-existence of a partition into plane spanning trees (Theorem 16), and give a complete characterization for partitions into plane double stars (Theorem 17). As both results need more notation, their statements are deferred to their section (the same holds for the remaining results).

Given the negative answers to Questions 1 and 2, a natural generalization is to study partitions into beyond planar subgraphs, that is, subgraphs in which certain restricted crossing patterns are allowed. We initiate this study for two important classes of beyond planar graphs, namely, *k-planar subgraphs* (where every edge is crossed by at most k other edges) and *k-quasi-planar subgraphs* (in which no k edges pairwise cross). For the former, we show bounds on the number of subgraphs required for partitioning $K(P)$ for P in convex position (Proposition 19 and Theorem 20). For the latter, we show that a partition into 3-quasi-planar spanning trees is possible for any P with $|P|$ even (Lemma 23). This is best possible, as 2-quasi-planar graphs are plane. We further present bounds on the partition of any $K(P)$ into k -quasi-planar subgraphs for general k (Theorem 25).

We remark that it is straightforward to model the problem of partitioning into (plane) subgraphs as an integer linear program (ILP), which easily computes solutions for point sets up to roughly 25 points. None of the proofs in this paper rely on the computer assisted ILP, but it served as a great source of inspiration (see the full version [8] for further details).

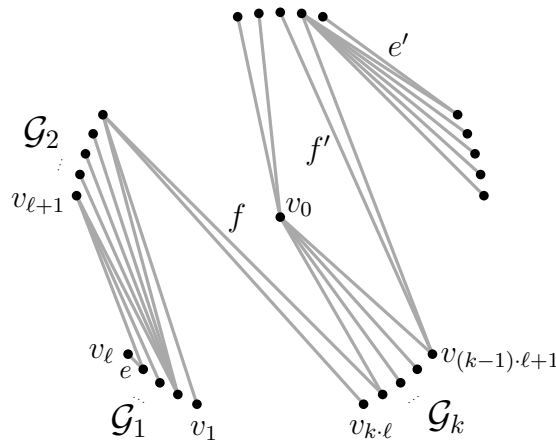
Organization of the paper. In Section 2, we prove Theorem 3 and Theorem 4, where we focus on the part showing the non-existence of partitions. In Section 3, we generalize our ideas from Section 2 about regular bumpy wheels to general wheel sets, proving Theorem 16 and Theorem 17. Finally, Sections 4 and 5 are dedicated to the more general setting of partitioning into k -planar and k -quasi-planar subgraphs, respectively.

2 Bumpy wheels

For a graph in (bumpy) wheel configuration we denote the center vertex by v_0 and the remaining vertices by v_1, \dots, v_{2n-1} in clockwise order. We also enumerate the groups in clockwise order: for $i \in \{1, \dots, k\}$, \mathcal{G}_i denotes the i 'th group (\mathcal{G}_1 contains v_1 , \mathcal{G}_k contains v_{2n-1})². An edge having v_0 as an endpoint is called a *radial edge*, an edge on the convex hull is called a *boundary edge* and all other edges are called *diagonal edges*. For a non-radial edge e , we define e^- to be the open halfplane defined by (the supporting line through) e and not containing v_0 , and similarly e^+ to be the open halfplane containing v_0 .

Additionally, we define a partial order $<_c$ on the set of non-radial edges, where $e <_c f$ if (the relative interior of) e completely lies in f^- (that is, f is “closer” to the center vertex v_0 than e). Two non-radial edges e, f are *incomparable* with respect to $<_c$, if neither $e <_c f$ nor $f <_c e$ holds (we omit “with respect to $<_c$ ” if it is clear from the context). In the following, when speaking of an edge e lying in f^- or in f^+ for another edge f , we always refer to the relative interior of e (that is, an endpoint of e may lie on the line through f – which actually means to coincide with an endpoint of f). A non-radial edge e is *maximal* in some set of edges E , if there is no other edge $e' \in E$ such that $e <_c e'$ (in the following we often consider maximal diagonal edges of plane spanning trees). *Minimal* edges are defined similarly. See Figure 2 for an illustration. Let us emphasize that we never use $<_c$ for radial edges.

² We will consider the index of a group \mathcal{G}_x always modulo k , but tacitly mean $((x-1) \bmod k) + 1$ (since our indexing starts with 1). The same holds for any other objects, e.g., the vertices on the convex hull.



■ **Figure 2** Example of a plane spanning tree on the bumpy wheel set $BW_{5,5}$. The diagonal edges f and f' are maximal. The edges e and e' are boundary edges (they are also the only minimal edges).

2.1 Partition into plane spanning trees

In this section, we prove Theorem 3. We remark that the non-existence direction almost follows from Theorem 4 (not even a partition into plane subgraphs is possible). The only case that is not covered is $BW_{3,5}$, which one can easily verify using computer assistance. However, since the proof of Theorem 3 is more instructive and intuitive, we decided to present it anyway and limit the proof of Theorem 4 to the essentials. We start with the non-existence:

► **Theorem 5.** *For any odd parameters $k \geq 3$ and $\ell \geq 5$, the edges of $BW_{k,\ell}$ cannot be partitioned into $n = \frac{k\ell+1}{2}$ plane spanning trees.*

Towards the proof of Theorem 5, we will first prove several structural results concerning the number and arrangement of radial and diagonal edges in the spanning trees of a potential partition (some of which have a similar flavor as those by Trao et al. [14]). We show that radial edges must lie between maximal diagonal edges and those maximal diagonal edges need to fulfill certain distance constraints. We will show that this cannot be satisfied if $\ell \geq 5$. Due to space constraints, we postpone the proofs of most preliminary results to the full version of this paper [8].

The following observation follows immediately from the construction of bumpy wheel sets and the definition of the partial order $<_c$.

► **Observation 6.** *For two non-radial, non-crossing, incomparable edges e, f the vertices in e^- and f^- are disjoint and neither e^- nor f^- contains an endpoint of the other edge.*

Note that e and f in the above observation may share an endpoint. Furthermore, for any set of edges E , two maximal edges $e, e' \in E$ are always incomparable.

► **Lemma 7.** *Let T be a plane spanning tree of $BW_{k,\ell}$. Then the following properties hold:*

- (i) *for any diagonal edge $e \in E(T)$, T contains at least one boundary edge in e^- ,*
- (ii) *for any pair of incomparable diagonal edges $e, f \in E(T)$, the boundary edges of T in e^- and f^- are distinct, and*
- (iii) *if T contains exactly one maximal diagonal edge, T contains at least $(\frac{k-1}{2}\ell + 1)$ consecutive radial edges (in particular, all radial edges of $\frac{k-1}{2}$ consecutive groups).*

6:6 Edge Partitions of Complete Geometric Graphs

Note that any spanning tree in a partition of $BW_{k,\ell}$ contains a maximal diagonal edge, since the star around v_0 clearly cannot be used in such a partition.

► **Proposition 8.** *Let T_0, \dots, T_{n-1} be a partition of $BW_{k,\ell}$ into plane spanning trees (if it exists). Then exactly one of those trees, say T_0 , contains a single boundary edge and a single maximal diagonal edge and all other $n - 1$ trees contain exactly two boundary edges and exactly two maximal diagonal edges each. In particular, any diagonal edge $e \in E(T_i)$ contains exactly one boundary edge of T_i in e^- .*

From now on, T_0 always denotes the spanning tree with exactly one boundary edge (when considering a partition into plane spanning trees). Further, we let all radial edges $\{v_0, v_i\}$ for $i \in \{1, 2, \dots, \frac{k-1}{2}\ell + 1\}$ be part of T_0 (which we can assume without loss of generality due to symmetry).

For two non-radial, non-crossing edges e, f , define the *span* of e and f to be the (closed) area between the two edges, that is,

$$\text{span}(e, f) = \begin{cases} \text{cl}(e^+ \cap f^+) & \text{if } e \text{ and } f \text{ are incomparable} \\ \text{cl}(e^+ \cap f^-) & \text{if } e <_c f, \end{cases}$$

where $\text{cl}(\cdot)$ denotes the closure. The shaded area in Figure 3 for instance defines the span of two incomparable edges e and f .

Note, however, that we are more interested in the vertices and edges contained in the span, rather than the area itself. If we want to emphasize this, we may use the notation $V(\text{span}(e, f))$ or $E(\text{span}(e, f))$. In the following we are mostly interested in the span of maximal diagonal edges of some plane spanning tree.

► **Lemma 9.** *Let T_0, \dots, T_{n-1} be a partition of $BW_{k,\ell}$ into plane spanning trees (if it exists) and e, f be the maximal diagonal edges of some T_i ($i \neq 0$). Then, all edges of T_i in the span of e and f are radial (except e and f).*

Define the *distance* $\text{dist}(e)$ of a non-radial edge e to be the number of vertices in e^- plus one (or in other words, the number of boundary edges in $\text{cl}(e^-)$). Clearly, $1 \leq \text{dist}(e) \leq \frac{k+1}{2}\ell - 1$ holds for any non-radial edge e and $\text{dist}(f) < \text{dist}(e)$ holds for any edge $f \subseteq e^-$. It will be convenient to define, for $i \in \{1, \dots, \frac{k+1}{2}\ell - 1\}$:

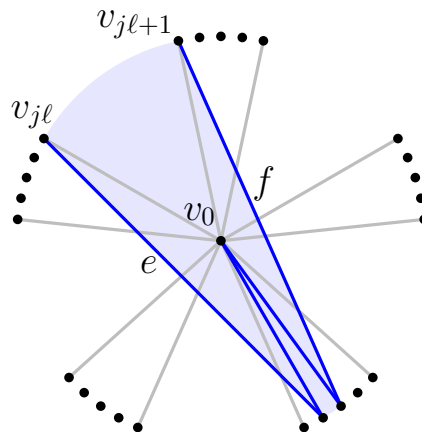
$$d_i = \frac{k+1}{2}\ell - i. \tag{1}$$

We define it in this (slightly counter-intuitive) way, d_1 being the largest distance, since we mostly deal with edges of large distances and thereby aim to improve the readability.

► **Lemma 10.** *Consider a plane spanning tree T of a partition of $BW_{k,\ell}$ and let e be a diagonal edge in T of distance $d = \text{dist}(e) > 1$. Then T also contains exactly one of the edges of distance $d - 1$ in e^- .*

We need a little more preparation towards the proof of Theorem 5. We call the first and last vertex of each group *outmost vertices* (and the corresponding radial edges *outmost radial edges*). Note that there are exactly $2k$ outmost radial edges in $BW_{k,\ell}$. Every hull vertex or radial edge that is not outmost, is called an *inside* vertex or an *inside* radial edge.

Furthermore, define two groups $\mathcal{G}_i, \mathcal{G}_j$ to be *opposite* if $|i - j| = \frac{k-1}{2}$ or $|i - j| = \frac{k+1}{2}$. In particular, each group has two opposite groups and two consecutive groups have exactly one opposite group in common (we call that group *the opposite group* of a pair of consecutive groups).



■ **Figure 3** All outmost radial edges are depicted in gray. The maximal diagonal edges e and f (connecting opposite groups) form a special wedge. Their span is shaded blue.

Let e, f be two maximal (non-crossing) diagonal edges which have an endpoint in a common group. Then the set of vertices of $\text{span}(e, f)$ in the common group is called *apex*. Note that any apex contains at least one vertex (and this lower bound is attained if the endpoints of e and f coincide).

Moreover, two maximal (non-crossing) diagonal edges $e = \{u, v\}$ and $f = \{u', v'\}$ form a *special wedge* if two endpoints (say u and u') are consecutive outmost vertices of different groups (that is, $u = v_{j\ell}$ and $u' = v_{j\ell+1}$ for some j) and v and v' are inside vertices lying in the opposite group of \mathcal{G}_j and \mathcal{G}_{j+1} . See Figure 3 for an illustration of these terms.

► **Proposition 11.** *Let T_0, \dots, T_{n-1} be a partition of $BW_{k,\ell}$ into plane spanning trees (if it exists) and let T_i ($i \neq 0$) be a spanning tree that does not use any outmost radial edge. Then the two maximal diagonal edges e, f of T_i form a special wedge and T_i has to use all radial edges incident to the apex of this wedge.*

Proof. We first argue that all but exactly two radial edges in $\text{span}(e, f)$ must be part of T_i . The subgraph of T_i induced by $V(\text{span}(e, f))$ needs to form a tree. Moreover, $\text{span}(e, f)$ contains $|V(\text{span}(e, f))| - 1$ radial edges. Since T_i uses the two diagonal edges $e, f \in E(\text{span}(e, f))$ and all other edges in the span need to be radial (Lemma 9), it has to use exactly all but two radial edges.

Furthermore, since we cannot have two maximal diagonal edges between the same pair of groups, the span of e and f contains at least two outmost vertices, namely in two different groups which contain an endpoint of e and f , respectively. On the other hand, $\text{span}(e, f)$ can neither contain a third outmost vertex nor an outmost vertex in its interior, since otherwise T_i has to use an outmost radial edge (by Lemma 9 and above argument). In particular, e and f share a common group and the apex does not contain any outmost vertex (hence, e and f form a special wedge, as depicted in Figure 3).

Moreover, since T_i has to use all but two radial edges in the span, it clearly has to use all radial edges incident to the apex. ◀

Note that for two spanning trees T_i, T_j ($i \neq j$) not using an outmost radial edge, their apices must be disjoint.

6:8 Edge Partitions of Complete Geometric Graphs

► **Proposition 12.** *Let T_0, \dots, T_{n-1} be a partition of $BW_{k,\ell}$ into plane spanning trees (if it exists). Then for each pair $\mathcal{G}, \mathcal{G}'$ of opposite groups and each $j \in \{1, \dots, \ell\}$ there is a unique diagonal edge (connecting \mathcal{G} and \mathcal{G}') of distance d_j (recall Equation (1)) that is maximal in its tree.*

Proof. Observe first that for any $j \in \{1, \dots, \ell\}$ there are exactly j edges of distance d_j (between \mathcal{G} and \mathcal{G}') and all edges of the same distance (between \mathcal{G} and \mathcal{G}') pairwise cross. Also note, for any two edges e, e' (between \mathcal{G} and \mathcal{G}') with $\text{dist}(e) > \text{dist}(e')$, either $e' \subseteq e^-$ holds or they cross. In particular, if they do not cross and belong to the same tree, the shorter is not a maximal edge.

Consider now for some $j \in \{2, \dots, \ell\}$ the distance d_j and let c_1, \dots, c_j be the colors³ used for all edges of this distance. By Lemma 10, there are $j - 1$ edges of (the larger) distance d_{j-1} using the same color as an edge of distance d_j , w.l.o.g. c_1, \dots, c_{j-1} . By the above arguments the corresponding edges of distance d_j cannot be maximal.

On the other hand, the color c_j cannot be used by any edge of larger distance, since again by Lemma 10 this color would have to appear in d_{j-1} as well. Hence, indeed the only edge of distance d_j that is maximal in its tree is the one of color c_j .

Lastly, for $j = 1$ observe that the single edge of distance d_1 is clearly maximal. ◀

Finally, we are ready to prove Theorem 5, which we restate here for the ease of readability:

► **Theorem 5.** *For any odd parameters $k \geq 3$ and $\ell \geq 5$, the edges of $BW_{k,\ell}$ cannot be partitioned into $n = \frac{k\ell+1}{2}$ plane spanning trees.*

Proof. Assume to the contrary that there is such a partition T_0, \dots, T_{n-1} . There are $2k$ outmost radial edges and T_0 uses (at least) k of them (see the remark after Proposition 8). Hence, there are at most $k + 1$ spanning trees (including T_0) containing an outmost radial edge.

Next, let us count how many spanning trees *not* containing an outmost radial edge we can have. Since, by Proposition 11, the apex of such a tree cannot use any outmost vertex nor any vertex already incident to a radial edge in T_0 , there remain $\frac{k+1}{2}(\ell - 2)$ possible vertices (to be used by apexes), namely the inside vertices of the last $\frac{k+1}{2}$ groups $\mathcal{G}_{\frac{k+1}{2}}, \dots, \mathcal{G}_k$ (which are not fully connected to v_0 by radial edges in T_0). Also recall that each apex contains at least one vertex.

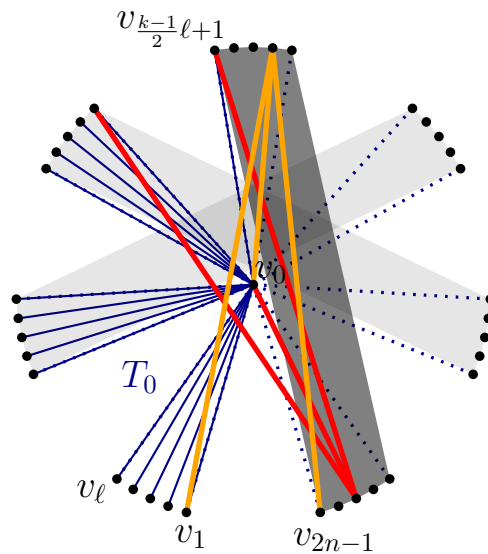
It is crucial to emphasize that among those last $\frac{k+1}{2}$ groups, group $\mathcal{G}_{\frac{k+1}{2}}$ and group \mathcal{G}_k are opposite (the only opposite pair). Therefore, by Proposition 11, two spanning trees with an apex in group $\mathcal{G}_{\frac{k+1}{2}}$ and group \mathcal{G}_k respectively, must each have a maximal diagonal edge between these two groups. Hence, by Proposition 12, we can have at most $(\ell - 2)$ spanning trees with apex in one of these two groups (instead of $2(\ell - 2)$); see Figure 4.

So, in total there can be at most $\frac{k-1}{2}(\ell - 2)$ spanning trees which do not use an outmost radial edge. Hence, whenever

$$k + 1 + \frac{k-1}{2}(\ell - 2) < \frac{k\ell + 1}{2}$$

holds, we cannot find enough spanning trees. Rearranging terms, this inequality is equivalent to $\ell > 3$. ◀

³ Instead of always spelling out that an edge belongs to a plane subgraph, we associate edges with colors.



■ **Figure 4** In the black stripes (the darker one is the crucial one) the maximal diagonal edges (of those trees without outmost radial edge) need to have distinct distances. That allows $\ell - 2$ many for each stripe. Two spanning trees (red and orange) with apex in group $\mathcal{G}_{\frac{k+1}{2}}$ and group \mathcal{G}_k both need to have a maximal diagonal edge in the dark stripe.

Next, we prove the other direction of Theorem 3:

► **Theorem 13.** *For any odd parameter $k \geq 3$, the edges of $BW_{k,3}$ can be partitioned into plane spanning trees.*

We only sketch the construction very briefly (the details can be found in the full version [8]).

Proof sketch. Our construction consists of three steps. In the first step, we give an explicit construction of a *partial partition* that covers all radial edges, each (partial) tree in the partition covers exactly its span, and between any pair of opposite groups exactly one diagonal edge of each distance d_1, d_2, d_3 is covered.

After that we extend this partial partition in two steps (these extensions actually work for arbitrary ℓ , but we stick to $\ell = 3$ for now). First we show that there is a unique way to extend the partial partition to one that covers all diagonal edges of distance d_1, \dots, d_3 . Roughly speaking, whenever we want to include some edge of distance d_i (between a certain pair of groups) we have two choices to which tree we can join it (see Lemma 10). However, since by construction exactly one edge of each distance is already covered, this determines the *orientation* how we can include the other edges of the same distance.

Once we covered all edges down to distance d_3 , there are precisely $2n - 1$ edges of each following distance and no edge of any smaller distance is already covered. Therefore, in each iteration (considering some distance $d_j < d_3$) we have the choice to fix some orientation (“left” or “right”) which determines how we need to extend all edges of distance d_j . Hence, in this second extension step there are $2^{\frac{3(k-1)}{2}-1}$ possible extensions. ◀

2.2 Partition into plane subgraphs

In the previous section, we gave a classification of which bumpy wheels can be partitioned into plane spanning trees and which cannot. Surprisingly it turns out that allowing arbitrary plane subgraphs does not help much. The only bumpy wheel that can be partitioned into plane subgraphs but not into plane spanning trees is $BW_{3,5}$.

Note that before we also heavily exploited the structure enforced by spanning trees. This is not possible anymore for the case of arbitrary plane subgraphs. We cannot make any assumptions on the number of edges, not even about connectedness. The only property we can (and will) exploit is the fact that we still have maximal diagonal edges and radial edges may only be contained in their span.

We split the proof of Theorem 4 into two parts, first focusing on the case $\ell > 5$.

► **Theorem 14.** *For any odd parameters $k \geq 3$ and $\ell > 5$, the edges of $BW_{k,\ell}$ cannot be partitioned into $n = \frac{k\ell+1}{2}$ plane subgraphs.*

The proof is more technical than for spanning trees. We give a detailed overview of the main ideas and postpone the full proof to the full version [8].

Proof sketch. Assume D_0, \dots, D_{n-1} is a partition into plane subgraphs. Then, a crucial insight is that between any pair of opposite groups and any distance $d_i = \frac{k+1}{2}\ell - i$ (for $1 \leq i \leq \ell$) there have to be at least i diagonal edges of distance at least d_i which are maximal in their subgraph. This follows from the fact that all edges of distance d_i between a fixed pair of opposite groups $\mathcal{G}, \mathcal{G}'$ form a crossing family. In particular, all of them get a different color and are either maximal or have another (larger) maximal edge between \mathcal{G} and \mathcal{G}' .

Further, this enables us to define a set E_{forced} of exactly $k \cdot \ell$ forced diagonal edges fulfilling the just mentioned distance constraints. In particular,

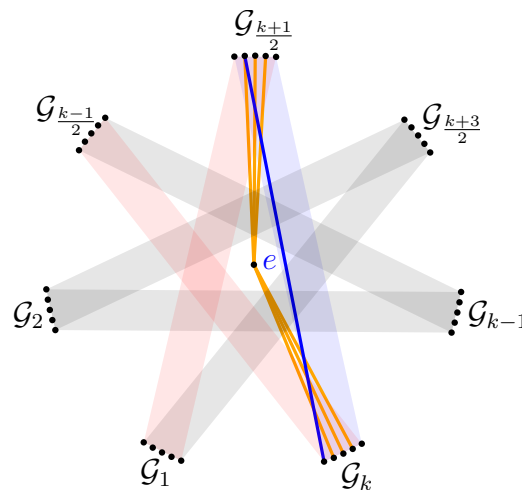
$$\sum_{e \in E_{\text{forced}}} \text{dist}(e) \geq k \sum_{i=1}^{\ell} \left(\frac{k+1}{2}\ell - i \right)$$

holds. Our goal will be to argue that we cannot accommodate all these forced diagonal edges and all radial edges at the same time.

To this end, note that we cannot have too many pairwise incomparable edges in a plane subgraph, more precisely their distance sums to at most $2n - 2$. In fact, it turns out that again we have one subgraph, say D_0 , containing exactly one forced diagonal edge, while all other $n - 1$ subgraphs contain exactly two of them.

Now the pairs of forced diagonal edges in our subgraphs again form a *span* (similar as in the spanning tree setting). Furthermore, radial edges may only be contained in this span (be careful, we are not assuming that there are radial edges in the span, but if the subgraph wants to use a radial edge it has to be in the span). We noted above that the distances of forced diagonal edges in the subgraph D_i sum up to at most $2n - 2$, say they sum up to $2n - 2 - x_i$ (and $\text{dist}(e) = d_1 - x_0$ for the single forced diagonal edge e of D_0). Then these x_i 's allow some additional margin to accommodate radial edges in the spans (or *additional vertices* as we call them). However, and this is the second crucial insight, we show that this additional margin is at most

$$\sum_{i=0}^{n-1} x_i \leq \frac{\ell - 1}{2}.$$



■ **Figure 5** High level overview of the proof of Theorem 15. We have at most $\frac{5-1}{2} = 2$ additional vertices in total and the blue stripe (which contains the single forced diagonal edge e of D_0) has to use both of them. Then, in the grey stripes we must use all forced diagonal edges of distances d_2, d_3, d_4 . Finally, since the two red stripes intersect ($k \geq 5$), there will not be enough forced diagonal edges left to pair all 6 forced diagonal edges of distances d_2, d_3, d_4 from the red stripes.

Finally, we consider only the $2\ell - 4$ inside radial edges of the opposite pair of groups $\mathcal{G}, \mathcal{G}'$ containing the endpoints of e (the single forced diagonal edge of D_0). Any subgraph with an apex in one of the two groups also has a forced diagonal edge between them. Putting everything together, this implies that we can cover at most

$$(\ell - 1) + \frac{\ell - 1}{2} = \frac{3}{2}(\ell - 1)$$

of these $2\ell - 4$ inside radial edges. In other words, whenever $\frac{3}{2}(\ell - 1) < 2\ell - 4$ holds, we cannot cover all edges. This inequality is equivalent to $\ell > 5$. ◀

For the case $\ell = 5$, we need to go even deeper into the structure of our plane subgraphs.

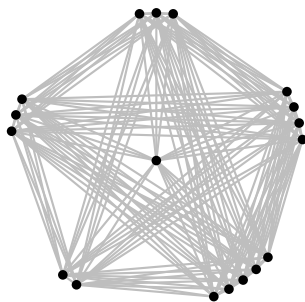
► **Theorem 15.** *For any odd parameter $k \geq 5$, the edges of $BW_{k,5}$ cannot be partitioned into $n = \frac{5k+1}{2}$ plane subgraphs.*

Figure 5 gives a brief sketch of the proof from a high level view. The full proof can also be found in the full version [8].

Finally, using Theorem 3, it only remains to show that there is a partition for $BW_{3,5}$, which is easy to compute (using computer assistance), and can be found in the full version [8].

3 Generalized wheels

In this section we generalize our construction to non-regular wheel sets. We give a sufficient condition in the setting of plane spanning trees and a full characterization for partitioning into plane double stars. For $N = [n_1, \dots, n_k]$ and integers $n_i \geq 1$, GW_N denotes the *generalized wheel* with group sizes n_i (in the given circular order). As before, the arrangement of the k groups resembles a regular k -gon around the center vertex, the vertices within each group are ε -close, and k is odd (see Figure 6). And for our purpose we also require $\sum_i n_i$ to be odd.



■ **Figure 6** Illustration of a generalized wheel ($GW_{[2,3,3,4,5]}$).

Note that the geometric regularity of generalized wheels is not strictly required (but eases the proofs). In fact, one can show that for any point set P (in general position) with exactly one point inside its convex hull, there is a generalized wheel with the exact same set of crossing edge pairs (further details can be found in the full version [8]).

► **Theorem 16.** *Let GW_N be a generalized wheel with k groups and $2n$ vertices. Then GW_N cannot be partitioned into plane spanning trees if each family of $\frac{k-1}{2}$ consecutive groups contains (strictly) less than $n - 2$ vertices.*

The proof, which is similar to the one of Theorem 5, can be found in the full version [8].

Plane double stars. Considering the other side of the story, it turns out that many generalized wheels can already be partitioned into plane double stars⁴:

► **Theorem 17.** *Let GW_N be a generalized wheel with k groups and $2n$ vertices. Then GW_N cannot be partitioned into plane spanning double stars if and only if there are three families of $\frac{k-1}{2}$ consecutive groups, each of which contains at most $n - 2$ vertices, such that each group is in at least one family.*

The proof requires several tools introduced by Schnider [13]. In a first step we identify conditions under which a point set admits a so-called *spine matching* – the collection of spine edges from a partition into double stars. Using these conditions we show that a generalized wheel GW_N cannot be partitioned into plane double stars if and only if GW_N has three *bad halfplanes* whose intersection is empty (for a non-radial *halving* edge e , the closure of e^- defines a *bad halfplane*). All details can be found in the full version [8].

We phrased Theorem 17 this way to make it consistent with Theorem 16; however, let us rephrase it in a way that better indicates the gap between the two theorems. Let F_i denote the family of $\frac{k-1}{2}$ consecutive groups starting at \mathcal{G}_i in clockwise order (whenever speaking of a family without further specification, we refer to such a family of $\frac{k-1}{2}$ groups for the remainder of this section). Two families F_i and F_{i+1} are called *consecutive* and $|F_i|$ denotes the number of vertices in F_i . If $|F_i| \leq n - 2$ holds, we call F_i *small*, and otherwise *large*.

► **Corollary 18.** *Let GW_N be a generalized wheel with k groups and $2n$ vertices. Then GW_N can be partitioned into plane spanning double stars if and only if there are $\frac{k-1}{2}$ consecutive families each containing (strictly) more than $n - 2$ vertices.*

⁴ All double stars in this section are spanning (which we may not always spell out for readability).

Proof. If, for the one direction, there are $\frac{k-1}{2}$ large consecutive families, then there is a group \mathcal{G}^* (namely the one that is contained in all these $\frac{k-1}{2}$ families) such that any family containing \mathcal{G}^* is large. In particular, there cannot be three small families covering all groups. Hence, by Theorem 17, there is a partition into plane double stars.

On the other hand, if there are no $\frac{k-1}{2}$ large consecutive families, we can find three small families as follows. Note first that every group is contained in some small family. Pick a small family F arbitrarily and let \mathcal{G} be the first group after F (in clockwise order). Among all small families containing \mathcal{G} , pick the one that is “furthest” from F , that is, has least overlap with F , and call it F' . Let \mathcal{G}' again be the first group after F' and among all small families containing \mathcal{G}' pick the one furthest from F' and call it F'' . Since F'' cannot contain \mathcal{G} , we conclude that the three small families F, F', F'' cover all groups. ◀

4 Partitions into k -planar subgraphs

In this section, we consider a generalization to partitioning into k -planar subgraphs (for $k = 0$ this amounts to the previous partitioning into plane subgraphs). We focus on the special case where the input point set is in convex position. Our first result fully resolves this problem for $k = 1$. Note that we do not require even sized point sets.

► **Proposition 19.** *For a point set P in convex position with $|P| = n \geq 5$, $K(P)$ can be partitioned into $\lceil \frac{n}{3} \rceil$ 1-planar subgraphs and $\lceil \frac{n}{3} \rceil$ subgraphs are required in every 1-planar partition.*

The proof can be found in the full version [3]. More generally, we show the following bounds:

► **Theorem 20.** *For an n -point set P in convex position and every $k \in \mathbb{N}$, $K(P)$ admits a partition into at most $\frac{n}{\sqrt{2k}}$ k -planar subgraphs. More precisely, for every integer $s \geq 2$, $K(P)$ admits a $\frac{(s-1)(s-2)}{2}$ -planar partition into $\lceil \frac{n}{s} \rceil$ subgraphs.*

Conversely, for every $k \in \mathbb{N}$, at least $\frac{n-1}{4.93\sqrt{k}}$ subgraphs are required in any k -planar partition of $K(P)$.

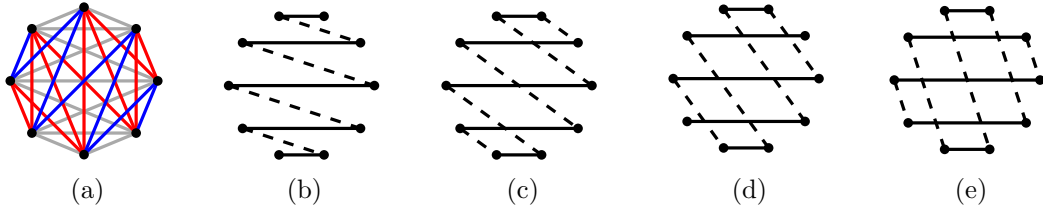
For the proofs of Proposition 19 and Theorem 20 (in particular for the lower bounds) it will be necessary to understand how many edges a single color class, or in other words, how many edges a k -planar subgraph of a convex geometric K_n , can maximally have. Once such bounds are established, we will be able to lower-bound the number of colors required in any k -planar partition of a convex geometric K_n by considering the “largest” color class.

We postpone this analysis, which also includes an improvement of the well-known crossing lemma for convex geometric graphs, to the full version [3] and only state the main ingredient that we need for the proof of Theorem 20:

► **Theorem 21.** *For every $k \geq 5$, every convex k -plane graph G on n vertices has at most $\sqrt{\frac{243}{40}k} \cdot n$ edges.*

Proof of Theorem 20. Let us first prove the upper bound. To this end, suppose that $s \geq 2$ is such that $\frac{(s-1)(s-2)}{2} \leq k$, and let us show that $K(P)$ can be partitioned into $\lceil \frac{n}{s} \rceil$ k -planar subgraphs. W.l.o.g. assume that the points in P form a regular n -gon. Consider all possible n slopes of segments and sort those in circular order. Next, partition this list of slope values into $\lceil \frac{n}{s} \rceil$ (contiguous) intervals of size at most s . Then, define a color class for all edges whose slopes fall into a common interval of this partition, see Figure 7(a).

6:14 Edge Partitions of Complete Geometric Graphs



■ **Figure 7** (a) Partition into 1-planar subgraphs by composing groups of (at most) 3 consecutive slopes each. (b)-(e) Edges with slope distance 1/2/3/4 intersect at most 0/1/2/3 times.

We show that all these subgraphs are $\frac{(s-1)(s-2)}{2}$ -planar. To this end, define the *slope distance* to be the distance between two slope values in the circularly sorted list of slopes. Note that edges cannot be crossed by other edges of the same slope or slope distance 1; by at most one edge of slope distance 2, by at most two edges of slope distance 3, etc. (see Figure 7(b)-(e)). Hence, if an edge e has color i , and if the slope of e is the j -th slope ($j \in \{1, \dots, s\}$) in its circular interval of slopes, then e can cross with at most the following amount of edges of color i :

$$\begin{aligned} \sum_{1 \leq k < j-1} (j-k-1) + \sum_{j+1 < k \leq s} (k-j-1) &= \frac{(j-1)(j-2)}{2} + \frac{(s-j)(s-j-1)}{2} = \\ &= \frac{(s-1)(s-2)}{2} - (s-j)(j-1) \leq \frac{(s-1)(s-2)}{2}. \end{aligned}$$

For the lower bound, note that $K(P)$ has $\frac{n(n-1)}{2}$ edges, and that in every k -planar partition of $K(P)$, every color class induces a convex k -plane subgraph on n vertices. Hence, by Theorem 21, every color class has size at most $\sqrt{\frac{243}{40}k} \cdot n$. So, the number of colors required in any k -planar partition is at least

$$\frac{\binom{n(n-1)}{2}}{\sqrt{\frac{243}{40}k} \cdot n} \geq \frac{n-1}{4.93\sqrt{k}}.$$

This concludes the proof. ◀

The following intriguing question is left open by our study.

► **Question 22.** *Is the upper bound in Theorem 20 tight up to lower-order terms?*

More generally, it would be interesting to shed some more light on the “in-between-cases” coming out of the upper bound in Theorem 20, where the term $\frac{(s-1)(s-2)}{2}$ covers only the values 0, 1, 3, 6, 10, For instance, can we partition convex complete geometric graphs with fewer colors into 2-planar subgraphs than we need for the 1-planar partition? More generally, for $\frac{(s-1)(s-2)}{2} < k < \frac{s(s-1)}{2}$, can we improve upon the $\lceil \frac{n}{s} \rceil$ bound from Theorem 20 for k -planar partitions? This question is surprisingly difficult (even for $k = 2$)⁵ and we do not know of any improvements of the bounds for these “in-between-cases”.

⁵ Using computer assistance, we can show that $\frac{3n}{10}$ colors are required for any 2-planar partition (almost matching the $\frac{n}{3}$ bound from the 1-planar partition). We omit this computer assisted result as it is a very special case and not even answering the question whether or not the bound can be improved for $k = 2$.

5 Partitions into k -quasi-planar subgraphs and spanning trees

In this section, we develop bounds on the number of colors required in a k -quasi-planar partition for point sets in general position (for $k = 2$ this again amounts to the setting of plane subgraphs, hence we assume $k \geq 3$ in the following). The setting of spanning trees is easily resolved by the following lemma (whose proof can be found in the full version [3]).

► **Lemma 23.** *Let P be a point set of size $2n$, then the complete geometric graph $K(P)$ can be partitioned into n 3-quasi-planar spanning trees.*

So, we turn our attention to the subgraph setting. The main ingredient towards the proof of Theorem 25 is the following lemma concerning point sets admitting a perfect crossing-matching, that is, a crossing family of size $|P|/2$. Note that in this case any edge in the crossing family determines a halving line [10].

► **Lemma 24.** *Let P be a point set of size $2n$, with a crossing family of size n , then $\lceil \frac{n}{k-1} \rceil$ colors are required and sufficient to partition $K(P)$ into k -quasi-planar subgraphs.*

Again, due to space constraints, we postpone the proof to the full version [3].

► **Theorem 25.** *Let P be a set of n points in general position and denote the size of a largest crossing family on P by m . Also let $k \geq 3$ s.t. $k \leq m$ (otherwise one color is always sufficient). Then, at least $\lceil \frac{m}{k-1} \rceil$ colors are required and at most $\lceil \frac{m}{k-1} \rceil + \lceil \frac{n-2m}{k-1} \rceil$ colors are needed to partition the complete geometric graph $K(P)$ into k -quasi-planar subgraphs.*

Proof. Let $P' \subseteq P$ be the subset of endpoints induced by a largest crossing family of size m .

Then, the lower bound follows immediately from Lemma 24 applied on P' .

For the upper bound, divide the point set $P \setminus P'$ into disjoint subsets Q_1, \dots, Q_c of size $k-1$, where $c = \lceil \frac{n-2m}{k-1} \rceil$. For each edge with an endpoint in some Q_i assign it the color i (for edges that have two choices, pick one arbitrarily). Certainly, each color class is k -quasi-planar, since it consists of (at most) the union of $k-1$ stars. Together with $K(P')$, which we can clearly partition by using $\lceil \frac{m}{k-1} \rceil$ colors, the upper bound follows. ◀

6 Conclusion

We showed that there are complete geometric graphs that cannot be partitioned into plane spanning trees and gave a full characterization of partitionability for bumpy wheels (even in the much broader setting of partitioning into plane subgraphs). Also, for generalized wheels we gave sufficient and necessary conditions. There are two natural directions for further research in this setting. On the one hand, one could try to further classify which point sets can be partitioned and which cannot (this might also be a useful approach towards the question concerning the complexity of the decision problem whether a given complete geometric graphs admits a partition into plane spanning trees). On the other hand, we initiated the study of partitions into broader classes of subgraphs, namely k -planar and k -quasi-planar.

The intriguing question to determine *how far* we may get from the $\frac{|P|}{2}$ bound is still open:

► **Question 26** ([6]). *Can any complete geometric graph on n vertices be partitioned into $\frac{n}{c}$ plane subgraphs for some constant $c > 1$?*

References

- 1 Oswin Aichholzer, Thomas Hackl, Matias Korman, Alexander Pilz, André van Renssen, Marcel Roeloffzen, Günter Rote, and Birgit Vogtenhuber. Packing plane spanning graphs with short edges in complete geometric graphs. *Comput. Geom.*, 82:1–15, 2019. doi:10.1016/j.comgeo.2019.04.001.
- 2 Oswin Aichholzer, Thomas Hackl, Matias Korman, Marc Van Kreveld, Maarten Löffler, Alexander Pilz, Bettina Speckmann, and Emo Welzl. Packing plane spanning trees and paths in complete geometric graphs. *Information Processing Letters*, 124:35–41, 2017.
- 3 Oswin Aichholzer, Johannes Obenaus, Joachim Orthaber, Rosna Paul, Patrick Schnider, Raphael Steiner, Tim Taubner, and Birgit Vogtenhuber. Edge Partitions of Complete Geometric Graphs (Part 2), 2021. arXiv:2112.08456.
- 4 Frank Bernhart and Paul C Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27(3):320–331, 1979. doi:10.1016/0095-8956(79)90021-2.
- 5 Ahmad Biniiaz and Alfredo García. Packing plane spanning trees into a point set. *Comput. Geom.*, 90:101653, 2020. doi:10.1016/j.comgeo.2020.101653.
- 6 Prosenjit Bose, Ferran Hurtado, Eduardo Rivera-Campo, and David R. Wood. Partitions of complete geometric graphs into plane trees. *Comput. Geom.*, 34(2):116–125, 2006. doi:10.1016/j.comgeo.2005.08.006.
- 7 <https://cgshop.ibr.cs.tu-bs.de/competition/cg-shop-2022/#problem-description>.
- 8 Johannes Obenaus and Joachim Orthaber. Edge partitions of complete geometric graphs (part 1), 2021. arXiv:2108.05159.
- 9 http://www.openproblemgarden.org/op/partition_of_complete_geometric_graph_into_plane_trees.
- 10 János Pach and József Solymosi. Halving lines and perfect cross-matchings. *Contemporary Mathematics*, 223:245–250, 1999.
- 11 Arkadiusz Pawlik, Jakub Kozik, Tomasz Krawczyk, Michal Lason, Piotr Micek, William T. Trotter, and Bartosz Walczak. Triangle-free intersection graphs of line segments with large chromatic number. *J. Comb. Theory, Ser. B*, 105:6–10, 2014. doi:10.1016/j.jctb.2013.11.001.
- 12 Patrick Schnider. Partitions and packings of complete geometric graphs with plane spanning double stars and paths. Master’s thesis, ETH Zürich, 2015.
- 13 Patrick Schnider. Packing plane spanning double stars into complete geometric graphs. In *Proc. 32nd European Workshop on Computational Geometry (EuroCG’16)*, pages 91–94, 2016.
- 14 Hazim Michman Trao, Gek L Chia, Niran Abbas Ali, and Adem Kilicman. On edge-partitioning of complete geometric graphs into plane trees. *arXiv preprint arXiv:1906.05598*, 2019.

Minimum-Error Triangulations for Sea Surface Reconstruction

Anna Arutyunova ✉

Institute for Computer Science,
Universität Bonn, Germany

Jan-Henrik Haunert

Institute of Geodesy and Geoinformation,
Universität Bonn, Germany

Jürgen Kusche

Institute of Geodesy and Geoinformation,
Universität Bonn, Germany

Philip Mayer ✉

Institute for Computer Science,
Universität Bonn, Germany

Heiko Röglin

Institute for Computer Science,
Universität Bonn, Germany

Anne Driemel

Hausdorff Center for Mathematics,
Universität Bonn, Germany

Herman Haverkort

Institute for Computer Science,
Universität Bonn, Germany

Elmar Langetepe

Institute for Computer Science,
Universität Bonn, Germany

Petra Mutzel

Institute for Computer Science,
Universität Bonn, Germany

Abstract

We apply state-of-the-art computational geometry methods to the problem of reconstructing a time-varying sea surface from tide gauge records. Our work builds on a recent article by Nitzke et al. (Computers & Geosciences, 157:104920, 2021) who have suggested to learn a triangulation D of a given set of tide gauge stations. The objective is to minimize the misfit of the piecewise linear surface induced by D to a reference surface that has been acquired with satellite altimetry. The authors restricted their search to k -order Delaunay (k -OD) triangulations and used an integer linear program in order to solve the resulting optimization problem.

In geometric terms, the input to our problem consists of two sets of points in \mathbb{R}^2 with elevations: a set \mathcal{S} that is to be triangulated, and a set \mathcal{R} of reference points. Intuitively, we define the error of a triangulation as the average vertical distance of a point in \mathcal{R} to the triangulated surface that is obtained by interpolating elevations of \mathcal{S} linearly in each triangle. Our goal is to find the triangulation of \mathcal{S} that has minimum error with respect to \mathcal{R} .

In our work, we prove that the minimum-error triangulation problem is NP-hard and cannot be approximated within any multiplicative factor in polynomial time unless $P = NP$. At the same time we show that the problem instances that occur in our application (considering sea level data from several hundreds of tide gauge stations worldwide) can be solved relatively fast using dynamic programming when restricted to k -OD triangulations for $k \leq 7$. In particular, instances for which the number of connected components of the so-called k -OD fixed-edge graph is small can be solved within few seconds.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Minimum-Error Triangulation, k -Order Delaunay Triangulations, Data dependent Triangulations, Sea Surface Reconstruction, fixed-Edge Graph

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.7

Related Version Full Version: <http://arxiv.org/abs/2203.07325>

Supplementary Material Software (Source Code and Information about the Data Acquisition):

<https://github.com/PhilipMayer94/dynamic-programming-for-min-error-triangulations>
archived at `swh:1:dir:a009b56de67b3679c496449aff63f6b343593a8d`



© Anna Arutyunova, Anne Driemel, Jan-Henrik Haunert, Herman Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel, and Heiko Röglin; licensed under Creative Commons License CC-BY 4.0

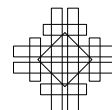
38th International Symposium on Computational Geometry (SoCG 2022).

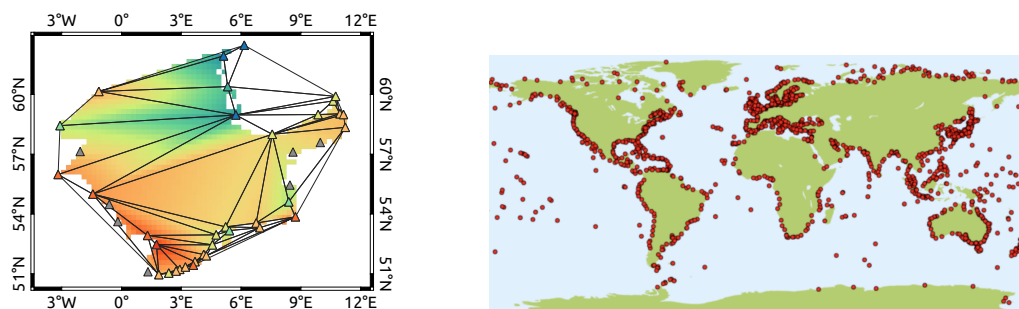
Editors: Xavier Goaoc and Michael Kerber; Article No. 7; pp. 7:1–7:18



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** Left: A minimum-error triangulation of the North Sea data (June 2010) with 34 tide gauge stations computed with the approach in [24]. Right: Locations of all tide gauge stations in the PSMSL database (www.psmsl.org/products/data_coverage).

Funding This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC-2047/1 – 390685813 and DFG grant RO 5439/1-1.

Acknowledgements We thank the anonymous reviewers for their insightful comments and suggestions.

1 Introduction

Reconstructing the sea level for the past is of paramount importance for understanding the influences of climate change. Two types of observational data are often used for this task: (1) data from tide gauge stations, which are usually located at the sea shore, and (2) gridded altimeter data acquired from satellites. The tide gauge data is available from the 18th century from stations that are sparsely distributed globally (e.g., the RLR database given by the PSMSL contains 1 548 stations). The gridded altimeter data, which has been acquired since 1993, admits much more accurate reconstructions of the sea surface for the last 29 years. We build on the work by Nitzke et al. [24], who suggested an approach for combining these two types of data using integer linear programming techniques. The approach is to learn a plausible triangulation of the tide gauge stations for an epoch E for which the altimeter data is available, and then use that triangulation to reconstruct the sea surface in another epoch, where gauge data is available, but no altimeter data. Given the gauge and altimeter data for E , the task is to compute a minimum-error triangulation of the gauge stations, that is, a triangulation that minimizes the sum of squared differences between the reference (altimeter) data and the piecewise linear surface defined with the triangulation.

For piecewise linear surfaces, Delaunay triangulations are often chosen, since they have many desirable properties. However, they are unique and so they do not have potential for optimization. On the other hand, computing a minimum-error triangulation among the set of all triangulations can lead to badly shaped triangles, which can cause large interpolation errors for epochs other than the training epoch. Therefore, Nitzke et al. [24] suggested computing a triangulation of minimum error among all k -order Delaunay (k -OD) triangulations [16]. A k -OD triangulation consists of triangles with up to k points inside each triangle’s circumcircle ($k = 0$ corresponds to Delaunay triangles). This creates room for optimization while ensuring (reasonably) well-shaped triangles. Moreover, restricting the solution to the set of k -order Delaunay triangulations has computational advantages. Nitzke et al. [24] modeled their approach as an integer linear program (ILP) and evaluated it on the North Sea dataset with up to 40 stations and $k \leq 3$, whose locations are projected on the

plane; see Figure 1. The evaluation showed that the k -OD minimum-error triangulation is substantially more effective than the method based on the Delaunay triangulation suggested in [25] for Sea Surface Anomaly reconstructions of up to 19 years back in time.

The aim of our work is to speed up the above approach using computational geometry in order to apply it to areas of global extent (instances with up to 800 tide gauge stations).

Our contribution.

- We first show that the minimum-error triangulation problem is NP-hard and that it is even NP-hard to approximate an optimal solution.
- We discuss an alternative optimization approach to the ILP-based one by Nitzke et al. [24]. Our approach is based on the dynamic programming (DP) algorithm by Silveira and Van Krefeld [28]. The runtime of the DP algorithm depends on the Delaunay order k ; since we are only interested in small orders, we are able to calculate minimum-error order- k Delaunay triangulations for the datasets given by the sea surface reconstruction problem.
- The algorithm's runtime depends on a subgraph of the Delaunay triangulation, which we call the order- k fixed-edge graph. It is known that for order 1 the fixed-edge graph is connected [16]. We investigate the fixed-edge graph for orders $k = 2, 3$. We show that for $k = 2$ no vertex can be isolated and give an example where the fixed-edge graph is not connected. For $k \geq 3$ we give an example where $\lfloor \frac{n}{6} \rfloor$ connected components are inside a face of the fixed-edge graph, which implies exponential runtime for the algorithm. This complements the observations by Silveira et al. given in [28].
- We perform experiments with different projections of the tide gauge dataset to analyze the structure of the fixed-edge graphs for a real-world dataset. Our experiments confirm the assumption by Silveira and Van Krefeld [28] that the DP algorithm can be used to solve practical problems for medium-sized datasets, if the order is small ($k \leq 7$).
- Lastly, we perform the reconstruction task that was given in [24] for the global dataset. Our evaluation shows that on the used global dataset with up to 800 stations the quality improves with growing k , which contrasts with the findings in [24] on the local North Sea dataset with about 40 stations, where $k = 2$ consistently delivered the best reconstructions.

The paper is organized as follows. First, we outline the formal definitions of the triangulation problem in Section 2. After that, we discuss related works in Section 3. In Section 4 we present our NP-hardness proof for the minimum-error triangulation problem. Section 5 presents the DP algorithm by Silveira et al. [28] and discusses our findings regarding the fixed-edge graphs. In Section 6 we provide the application of the DP algorithm to the sea surface reconstruction problem. Finally, we give our conclusion in Section 7.

2 The triangulation problem

Let $\mathcal{S} \subset \mathbb{R}^2$ be a set of n points and $f: \mathcal{S} \rightarrow \mathbb{R}$. We call \mathcal{S} the set of triangulation points and $f(s)$ the measurement value of $s \in \mathcal{S}$. Additionally, we are given a set $\mathcal{R} \subset \text{conv}(\mathcal{S})$ of m points and a function $h: \mathcal{R} \rightarrow \mathbb{R}$. We refer to \mathcal{R} as the set of reference points and to $h(r)$ as the reference value of $r \in \mathcal{R}$.

A triangulation D of \mathcal{S} is given by a maximal set of non-crossing straight-line edges between points in \mathcal{S} . We can extend the function f on the points in $\text{conv}(\mathcal{S})$ by linearly interpolating f in every triangle. In this way we obtain a piece-wise linear function $s_D: \text{conv}(\mathcal{S}) \rightarrow \mathbb{R}$. The *minimum-error triangulation problem* asks for a triangulation D of \mathcal{S} that minimizes the squared error between the reference values and the interpolation, i.e.,

$$\text{Err}_D(\mathcal{R}) = \sum_{r \in \mathcal{R}} (s_D(r) - h(r))^2.$$

For the dynamic programming algorithm used in our approach and discussed in Section 5, we transform the minimum-error triangulation problem to the *minimum triangle-weighted triangulation problem*. Let \mathfrak{T} be the set of all $O(n^3)$ possible triangles that may be used in any triangulation of \mathcal{S} . Then we can assign the weight

$$w_T(\mathcal{R}) = \sum_{r \in T} (s_T(r) - h(r))^2$$

to every triangle $T \in \mathfrak{T}$, where s_T is the linear interpolation given by the triangle T . If we assume that no reference point lies on any triangulation edge, we get

$$\text{Err}_D(\mathcal{R}) = \sum_{r \in \mathcal{R}} (s_D(r) - h(r))^2 = \sum_{T \in D} \sum_{r \in T} (s_T(r) - h(r))^2 = \sum_{T \in D} w_T(\mathcal{R}).$$

To get rid of the previous assumption we assign points that lie on an edge \overline{uv} only to the triangles left of \overrightarrow{uv} . Points coinciding with triangulation points can be ignored.

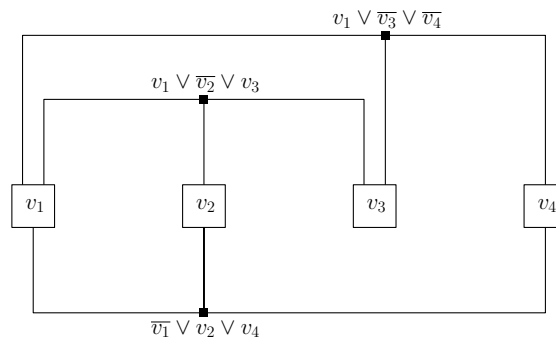
Using these weights our cost function becomes a decomposable measure as discussed by Bern and Eppstein in [6]. Broadly speaking, decomposable measures are all measures that easily allow computation using dynamic programming approaches for triangulations.

3 Related works

Sea level reconstruction. Conventional methods for sea level reconstruction use global base functions (empirical orthogonal base functions) which are *learned* within the altimeter decades [10]. Olivieri and Spada suggested the first triangulation-based reconstruction approach [25]. However, this approach does not use the altimeter data in any way and generates a Delaunay triangulation of the station data. Nevertheless, the resulting reconstruction of the sea surface was quite promising. The approach suggested by Nitzke et al. [24] marries the conventional thinking and the triangulation method. The authors proposed the use of data-dependent triangulations which were introduced in [12] by Dyn, Levin and Rippa. The particular focus of Nitzke et al. were the minimum-error triangulations. Since they also want to reconstruct the sea level in the pre-altimetry era, they formulate the reconstruction as a learning task and use higher-order Delaunay constraints, which were introduced in [16] by Gudmundsson, Hammar and van Kreveld, as regularizer.

Triangulating point sets. Triangulating point sets in the plane is a fundamental task of computational geometry. It is of high relevance for data interpolation and surface modeling tasks, where for every data point a data value (or height) is given in addition to the point's two coordinates. The Delaunay triangulation is most often applied as it optimizes several criteria and can be computed efficiently. In particular, it maximizes the minimum angle among all the angles of all the triangles. *Data-dependent triangulations* have been defined in [12] as triangulations that are computed under consideration of the data values. As optimization criteria the authors have considered (1) smoothness criteria, (2) criteria based on three-dimensional properties of the triangles, (3) variational criteria, and (4) the minimum-error criterion, which is optimized by the previously defined minimum-error triangulation.

There are many heuristics for computing data-dependent triangulations [3, 8, 12, 29], which are usually based on Lawson's edge flip algorithm [21]. For small instances, the problem can be solved to optimality based on integer linear programming [24]. There are multiple fixed-parameter-tractable algorithms using dynamic programming for the minimum-weight triangulation (MWT) problem [19, 9, 7, 4, 15] that can be adapted for decomposable measures [6]. Using problem specific structural properties the MWT problem has been solved for instances with up to 30 million points [17, 14].



■ **Figure 2** Embedding of the 3SAT formula $(\bar{v}_1 \vee v_2 \vee v_4) \wedge (v_1 \vee \bar{v}_2 \vee v_3) \wedge (v_1 \vee \bar{v}_3 \vee \bar{v}_4)$.

In [11, 27] heuristics and higher-order Delaunay constraints were used for terrain approximation. Using established techniques, exact polynomial-time algorithms can be obtained for restricted cases with higher-order Delaunay constraints [16, 28]. However, prior to our work, little was known about the complexity of computing or approximating minimum-error triangulations in the general case. For related problems some hardness results exist [2, 23].

4 Minimum-error triangulation is NP-hard

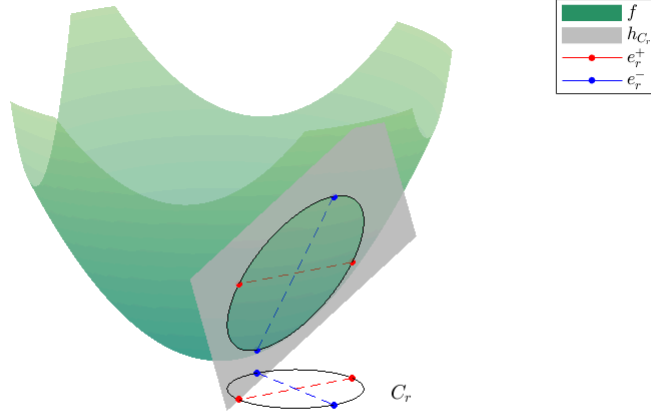
The *zero-error triangulation problem* asks for a triangulation D of \mathcal{S} with $s_D(r) = h(r)$ for all $r \in \mathcal{R}$, or equivalently $\text{Err}_D(\mathcal{R}) = 0$. We prove that this problem is NP-hard.

► **Theorem 1.** *The zero-error triangulation problem is NP-hard. Thus the minimum-error triangulation problem cannot be approximated within any multiplicative factor in polynomial time unless $P=NP$.*

We prove this by a reduction from the planar 3SAT problem, which is NP-complete [22]. An instance of this problem can be embedded into the plane, where every clause is represented by a vertex and every variable by a box placed on the horizontal axis. A box is connected to a vertex via a rectilinear edge if the respective variable is contained in the clause. For an example, see Figure 2. Such an embedding is also used, for example, in [20].

For every instance of the planar 3SAT problem we construct an instance for the zero-error triangulation problem by replacing the boxes, vertices and edges of its rectilinear embedding in the plane by a set of triangulation points and reference points. For this purpose we handle each component of the 3SAT embedding individually. We construct the *variable gadgets* which replace the boxes, the *wire gadgets*, which replace the rectilinear edges and finally the *clause gadgets* and the *negation gadgets*, where the first replace the vertices and the second can be attached to variable gadgets to handle negated variables in a clause. The combination of these gadgets then constitutes an instance to the zero-error triangulation problem.

We ensure that there are two possible zero-error triangulations on the points belonging to a variable gadget and the attached negation gadgets and wire gadgets as follows. Points from \mathcal{S} together with their measurement value can be seen as points in \mathbb{R}^3 . We ensure that they lie on a paraboloid in \mathbb{R}^3 and exploit the properties of the paraboloid (its convexity and the correspondence of planes in \mathbb{R}^3 to circles in \mathbb{R}^2) to limit possible zero-error triangulations. Any such triangulation then corresponds to the assignment of value 0 (negative) or 1 (positive) to any variable. We claim that the instance can be triangulated with zero error if and only if the 3SAT instance is solvable.



■ **Figure 3** Example of a reference point r with coupled circle C_r and its positive/negative edges crossing at r . Lifting the red and blue points to \mathbb{R}^3 , with their measurement values as third coordinate, we see that these points lie on both the paraboloid and the plane containing $(r, h_{C_r}(r))$.

4.1 Notation and local properties

Our triangulation instance consists of a set of triangulation points with integral coordinates $\mathcal{S} \subset \mathbb{Z}^2$ and a set $\mathcal{R} \subset \text{conv}(\mathcal{S})$ of reference points. The measurement value of a point $p = (p_1, p_2) \in \mathcal{S}$ is given by $f(p) = p_1^2 + p_2^2$. In contrast, reference values are not determined by one single function. Instead we define a set of functions, one for every circle in \mathbb{R}^2 , and choose for every reference point one of these functions which determines the reference value of this point. Concretely, let C be a circle around a point $x = (x_1, x_2)$ with radius ρ . We denote with $I_C = \{y \in \mathbb{R}^2 \mid \|x - y\|_2 < \rho\}$ the *interior* of C and with $O_C = \mathbb{R}^2 \setminus (C \cup I_C)$ the *exterior* of C . Here $\|\cdot\|_2$ denotes the Euclidean norm. For a reference point $r = (r_1, r_2) \in \mathcal{R}$ we define the function

$$h_C(r) = 2x_1r_1 + 2x_2r_2 - x_1^2 - x_2^2 + \rho^2.$$

The function graph of f is the unit paraboloid $\{(p_1, p_2, p_1^2 + p_2^2) \mid (p_1, p_2) \in \mathbb{R}^2\}$ and the function graph of h_C is the plane containing the lifting of C onto the paraboloid (Figure 3).

Every point $r \in \mathcal{R}$ is then *coupled* to a circle, which we denote by C_r . It will be defined during the construction of the gadgets and determines the reference value $h(r) = h_{C_r}(r)$. Let an edge $e = \overline{st}$ denote the convex hull of two points (its vertices) $s, t \in \mathbb{R}^2$. For each $r \in \mathcal{R}$ we define a *positive edge* e_r^+ and a *negative edge* e_r^- both having triangulation points lying on C_r as endpoints and intersecting each other at r (i.e., $e_r^+ \cap e_r^- = \{r\}$). Figure 3 shows the whole construction. We say for a triangulation D that the *signal* at $r \in \mathcal{R}$ is *positive* if D contains edge e_r^+ and *negative* if it contains e_r^- , otherwise we call it *ambiguous*. Similarly for every set $M \subset \mathcal{R}$ we call D *positive* on M if the signal at all $r \in M$ is positive and *negative* on M if the signal at all $r \in M$ is negative. The error incurred by D on M is given by

$$\text{Err}_D(M) = \sum_{r \in M} (s_D(r) - h(r))^2.$$

A triangle T is the convex hull of three points $s, t, u \in \mathbb{R}^2$, which we call the vertices of T . We say that a triangle T is in D if all of its edges $\overline{st}, \overline{tu}, \overline{us}$ are in D and T does not contain further points from \mathcal{S} , i.e., $T \cap \mathcal{S} = \{s, t, u\}$. We say that $r \in \mathcal{R}$ is *represented with zero error* by T if $r \in T$ and the value at r of the linear interpolation of f on T equals $h(r)$.

► **Lemma 2.** *Let r be a point of \mathcal{R} and let $T \subset \mathbb{R}^2$ be a triangle with vertices s, t, u and $r \in \text{conv}(\{s, t, u\} \cap C_r)$. Then r is represented with zero error by T .*

If the 3SAT instance is satisfiable, we argue that there is a triangulation containing one of e_r^\pm for every reference point r . Lemma 2 states that such a triangulation has in fact zero error (see also Figure 3). To represent r with zero error in any other way, we need at least one triangulation point inside and one outside C_r . This follows from the convexity of f .

► **Lemma 3.** *Let $T \subset \mathbb{R}^2$ be a triangle with vertices s, t, u representing $r \in \mathcal{R}$ with zero error. If $r \notin \text{conv}(\{s, t, u\} \cap C_r)$, then $\{s, t, u\}$ has a non-empty intersection with I_{C_r} and O_{C_r} .*

We guarantee during the construction that only few triangulation points lie in I_{C_r} for each reference point r . With a concise case analysis we rule out that any of them can be used together with a point in O_{C_r} to form a triangle that represents r with zero error, which limits the choice to triangles containing one of e_r^\pm . This ensures that every zero-error triangulation yields a solution to the 3SAT instance.

Our triangulation instance contains a set of *mandatory edges* that we require to be part of any feasible triangulation of \mathcal{S} . Mandatory edges are not part of the zero-error triangulation problem as defined in Section 2, but they can be eliminated by an additional construction.

4.2 The gadgets

At the core of our reduction lies the design of the gadgets that constitute the triangulation instance. Before we dedicate ourselves to the more complicated gadgets we construct smaller elements called *bits* and *segments* which then are combined into the larger gadgets.

A *bit* at $r \in \mathbb{Z}^2$ occupies a small construction around the central point r , which is also the only reference point of this bit, and can be oriented either horizontally or vertically. We describe the horizontal bit. Point r is coupled to a circle C_r which is centered on r and has radius $\sqrt{2}$. The integer grid points on this circle, that is, the points $r + (\pm 1, \pm 1)$, are triangulation points. Moreover $r + (0, 1)$ and $r + (0, -1)$ are triangulation points, whereas $r + (-2, 0)$, $r + (-1, 0)$, $r + (1, 0)$ and $r + (2, 0)$ are *not*. Therefore, we call the latter points *forbidden*. Furthermore we define the positive and negative edge as

$$e_r^+ = \text{conv}(r + (-1, -1), r + (1, 1)), \quad e_r^- = \text{conv}(r + (-1, 1), r + (1, -1)).$$

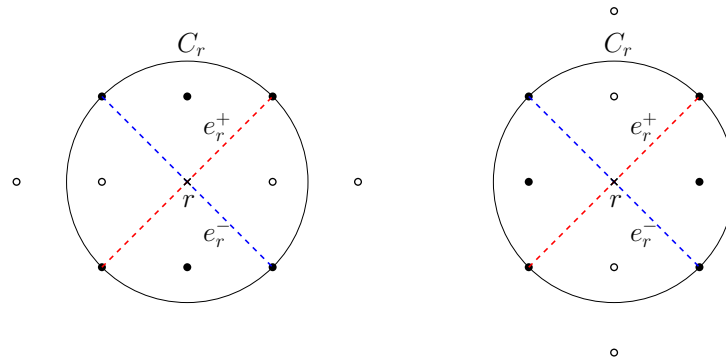
As $r + (\pm 1, \pm 1) \in C_r$, any triangle containing either e_r^+ or e_r^- represents r with zero error by Lemma 2. For the vertical bit we switch the definition of the positive and negative edge and rotate the whole construction by $\frac{\pi}{2}$. Figure 4 illustrates both constructions.

► **Lemma 4.** *Suppose the instance contains a bit at r . If $\mathcal{S} \subset \mathbb{Z}^2$ and \mathcal{S} does not contain forbidden points of the bit, any triangulation D of \mathcal{S} with $\text{Err}_D(r) = 0$ contains one of e_r^\pm .*

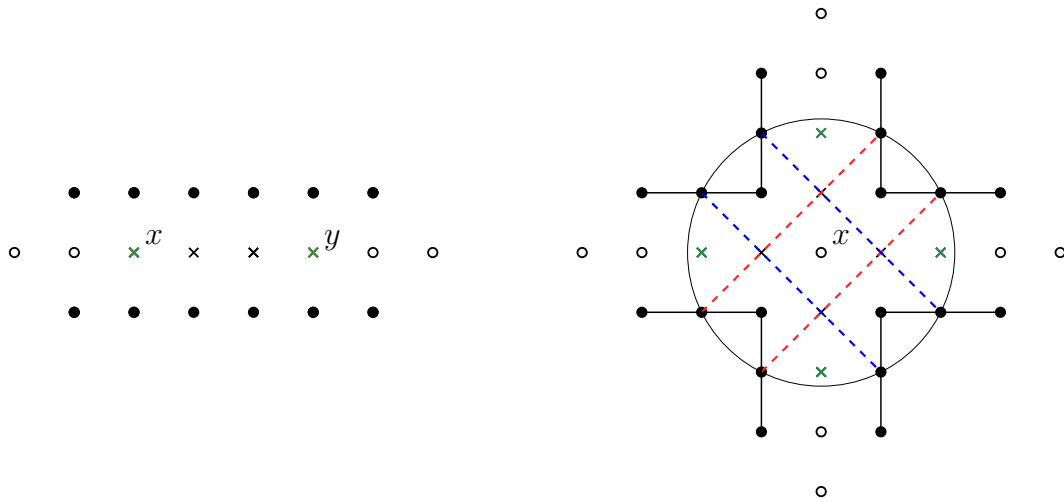
The next larger components are the *wire segment* and the *multiplier segment*, which we build from bits. They can be combined at specified reference points, which we call *anchor points*. These points are always reference points of bits.

A *wire segment* connects two points $x, y \in \mathbb{Z}^2$ lying on the same horizontal or vertical line. We place a horizontal or vertical bit on x, y and all integral points lying between these on the line connecting x and y . The anchor points of this segment are x, y .

A *multiplier segment* at a point $x \in \mathbb{Z}^2$ consist of two horizontal bits at $x \pm (2, 0)$ and two vertical bits at $x \pm (0, 2)$. These four points are simultaneously anchor points. Furthermore we add four inner reference points $x \pm (0, 1), x \pm (1, 0)$ whose coupled circle is of radius $\sqrt{5}$ and centered around x . So the circle contains the points $x + (\pm 2, \pm 1), x + (\pm 1, \pm 2)$. Figure 5



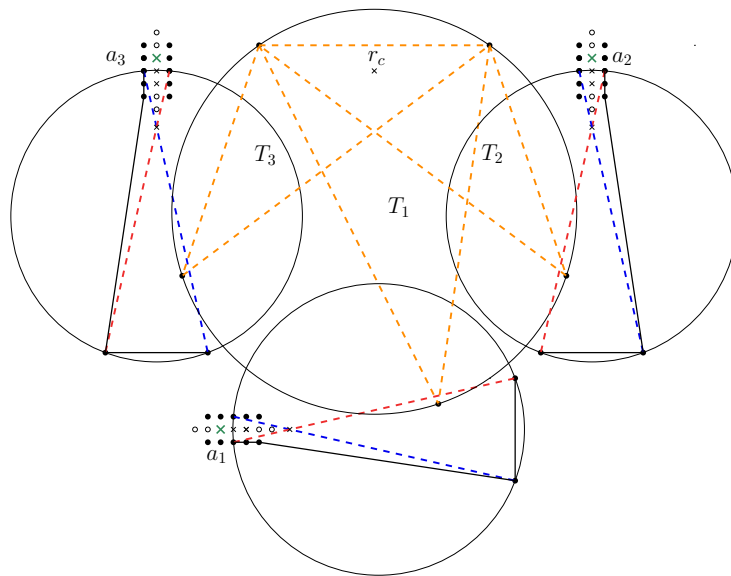
■ **Figure 4** The (horizontal/vertical) bit at r with the positive edge in red and the negative edge in blue. The black points are triangulation points and the white points are forbidden.



■ **Figure 5** Example of a horizontal wire segment on the left and a multiplier segment with mandatory edges on the right. The red or blue edges indicate the positive or negative edges of the crossing points, respectively. All white points and all reference points are forbidden. The green points are anchor points.

shows the wire segment and the multiplier segment including mandatory edges and the positive/negative edges of the inner reference points. To obtain the larger variable gadget and wire gadget we combine wire segments with multiplier segments. Two segments can be combined if they share a common anchor point. By the combination of two segments we mean the union of their reference points and triangulation points. A point is forbidden in the combination if it is forbidden in at least one of the segments. Thus it is not allowed to combine two segments if a triangulation point of one is forbidden in the other. The set of anchor points of the combination is defined as the symmetric difference of anchor point sets of both segments. This way we can combine arbitrarily many segments.

Remember that the *wire gadget* replaces the rectilinear edges of the 3SAT embedding, so it has to connect two points $x = (x_1, x_2), y = (y_1, y_2) \in \mathbb{Z}^2$. It consists of a multiplier segment placed on either (x_1, y_2) or (y_1, x_2) to form a corner, which is connected on two of its anchor points via two wire segments to both x and y . A *variable gadget* at $v \in \mathbb{Z}^2$ consists of ℓ multiplier segments at sufficiently large distance $\alpha \in \mathbb{Z}$, which we do not specify further. Here ℓ denotes the number of clauses. Concretely, we place a multiplier segment on each of



■ **Figure 6** The clause gadget, where the red/blue edges indicate the positive/negative edges of the crossing points. The triangles T_1, T_2, T_3 are orange and the anchor points a_1, a_2, a_3 green.

the points $v + (k\alpha, 0)$ with $0 \leq k \leq \ell - 1$ and connect them via horizontal wire segments at their anchor points. The multiplier segments ensure that the gadget can later be connected at its anchor points to multiple clause gadgets. We observe that the described combinations of segments for both gadgets are allowed and that they have the following crucial property.

► **Lemma 5.** *Suppose the instance contains a wire/variable gadget and let $\tilde{\mathcal{R}}$ be the reference points of this gadget. If $\mathcal{S} \subset \mathbb{Z}^2$ and \mathcal{S} does not contain forbidden points of the gadget, any triangulation D of \mathcal{S} with $\text{Err}_D(\tilde{\mathcal{R}}) = 0$ is either positive or negative on $\tilde{\mathcal{R}}$.*

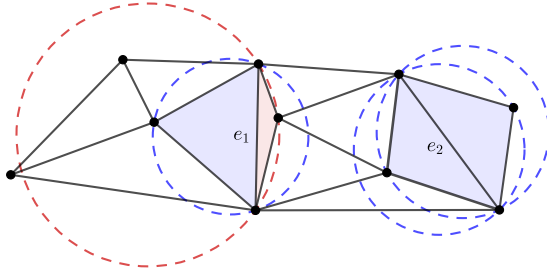
Now we define the *clause gadget* at a point $c \in \mathbb{Z}^2$, which combines three signals. To this end we add a reference point $r_c = c + (0, 11)$. Instead of a positive/negative edge it comes with three triangles T_1, T_2, T_3 whose vertices lie on C_{r_c} , each triangulating r_c with zero error. The clause gadget can be connected to other gadgets at three anchor points a_1, a_2, a_3 . With an additional construction we block the triangle T_i if the signal at a_i is positive for $i = 1, 2$ and T_3 if the signal at a_3 is negative. For the construction we refer to Figure 6 and [5].

► **Lemma 6.** *Suppose the instance contains a clause gadget and let $\tilde{\mathcal{R}}$ be its reference points. If $\mathcal{S} \subset \mathbb{Z}^2$ and \mathcal{S} does not contain forbidden points of the gadget, any triangulation D of \mathcal{S} with $\text{Err}_D(\tilde{\mathcal{R}}) = 0$ must be negative on one of the anchor points a_1, a_2 or positive on a_3 .*

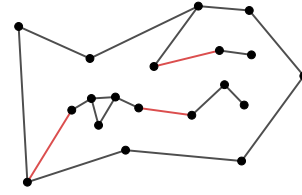
The last gadget, the *negation gadget*, is discussed in the full version [5]. It is constructed out of wires, multipliers and simplified clause gadgets. Finally, we replace the mandatory edges by an additional construction and argue that all gadgets keep their crucial properties. Using them we construct the zero-error triangulation instance and prove Theorem 1 in [5].

5 Higher-order Delaunay optimization

In the previous section we established that finding a minimum-error triangulation is NP-hard. Moreover, the experiments in [24] by Nitzke et al. suggest, that general minimum-error triangulations do not yield the most promising reconstructions of the sea surface. In their paper they used higher-order Delaunay (HOD) triangulations which allow a trade-off between a well shaped triangulation and a good approximation of the training dataset.



■ **Figure 7** A 2-OD triangulation; in blue the 1-OD and in red the 2-OD triangles; e_1 is a useful 2-OD edge and e_2 is a useful 1-OD edge.



■ **Figure 8** In black a (degenerate) polygon with connected components; in red one set H of connections.

In this section we summarize the algorithm given by Silveira et al. in [28]. Additionally, we extend upon their work by investigating the fixed-edge graphs in more detail.

We only consider point sets \mathcal{S} in general position, i.e., no four points lie on a circle and we denote the circle defined by three vertices $u, v, w \in \mathcal{S}$ by $C(u, v, w)$. A triangle $T_{u,v,w}$ is called an *order- k Delaunay (k -OD) triangle*, if $C(u, v, w)$ contains at most k points from \mathcal{S} in the interior. A triangulation is called *k -OD triangulation*, if all of its triangles have order k and an edge is called *useful k -OD edge*, if some k -OD triangulation of \mathcal{S} uses it; see Figure 7.

The minimum-error measure $\text{Err}_D(\mathcal{R})$ can be optimized using dynamic programming, since it is decomposable after pre-processing the triangle weights; see [6] for a formal definition. The well known DP algorithm that was independently proposed by Klincsek in [19] and Gilbert in [15] can be used to optimize polygon triangulations for decomposable measures in $O(n^3)$ time. In [28] the runtime of the DP algorithm is improved to $O(nk^2)$, if the algorithm only considers pre-processed k -OD edges and triangles instead of all possible ones.

Furthermore, Silveira et al. [28] extend the algorithm to the class of polygons P containing h connected components C_1, \dots, C_h ; see Figure 8. The algorithm performs an exhaustive search on a collection \mathcal{H} of sets of edges H , such that the planar graph $\bigcup_i C_i \cup P \cup H$ is connected for each $H \in \mathcal{H}$ and at least one H is used in the optimal triangulation. One of the main results in [28] is the existence of such a collection with size $O(k)^h$.

► **Theorem 7** (from [28]). *An optimal k -OD triangulation with respect to $\text{Err}_D(\mathcal{R})$ of a (degenerate) polygon with n boundary vertices and $h \geq 1$ components inside can be computed in $O(kn \log n) + O(k)^{h+2}n$ expected time.*

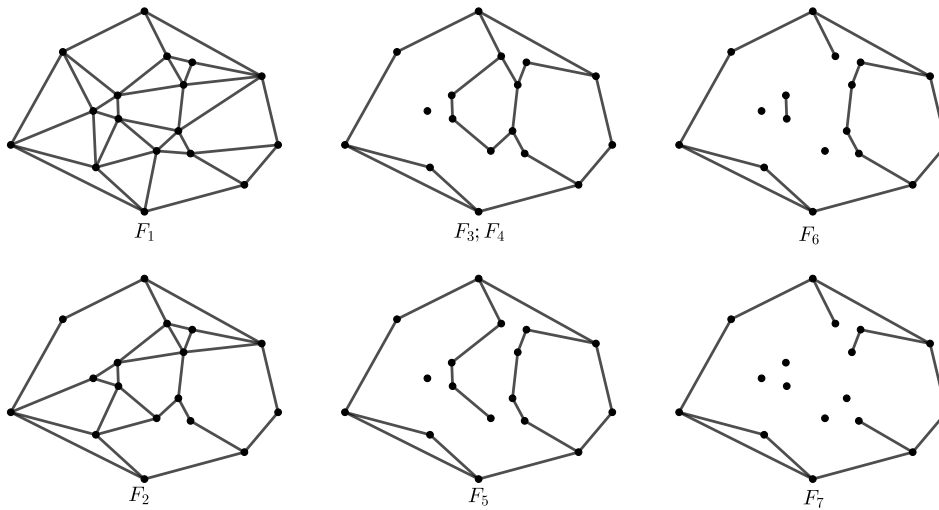
We can apply this algorithm to point sets by finding subgraphs F of the optimal triangulation [9, 28] and applying the DP algorithm to the faces of F .

5.1 The order- k fixed-edge graph

A subgraph that is naturally given by HOD constraints is the fixed-edge graph which was first discussed in [28]. The *order- k Delaunay (k -OD) fixed-edge graph* F_k of a pointset \mathcal{S} is given by all useful k -OD edges that are not intersected by any other useful k -OD edge.

► **Observation 8.** *Let \mathcal{S} be a set of n points. Let DT denote the Delaunay triangulation. We have $DT = F_0 \supset F_1 \supset F_2 \supset \dots \supset F_m = \dots = F_n \supset \text{conv}(\mathcal{S})$ for some $m \leq n$.*

In Figure 9 a sequence of fixed-edge graphs is illustrated. F_k decomposes the pointset into degenerate polygons P_1, \dots, P_m that may contain some connected components. An example is given in Figure 10. We can compute optimal solutions D_i for all P_i with the DP



■ **Figure 9** A sequence of fixed-edge graphs F_1, \dots, F_7 for an example point set.

algorithm. Since $\text{Err}_D(\mathcal{R})$ is decomposable, the optimal triangulation of \mathcal{S} is given by $\bigcup_i D_i$. Therefore, the runtime of the algorithm is dominated by the polygon with the maximum number of connected components c_{\max} . The application of Theorem 7 results in:

► **Corollary 9.** *An optimal k -OD triangulation of a point set \mathcal{S} with respect to $\text{Err}_D(\mathcal{R})$ can be computed in $O(kn \log n) + O(k)^{c_{\max}+2}n$ expected time.*

Next, we give some theoretical results with respect to the structure of F_2 and F_3 .

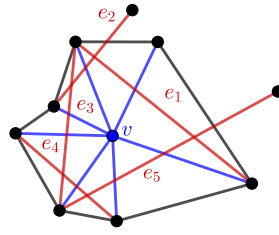
Let $v \in \mathcal{S}$ be a triangulation point. We call the graph N given by all edges of its incident Delaunay triangles its *Delaunay neighbourhood*, all of its incident edges in N its *connecting edges* and all other edges of N its *boundary edges*. A useful 2-OD edge that intersects a connecting edge is called *separation edge*; see Figure 11.

► **Theorem 10.** *Let \mathcal{S} be a set of points. Then every vertex in F_2 is adjacent to at least one other vertex of \mathcal{S} .*

Proof. (Sketch; the complete proof is given in the full version of the paper [5]) It is sufficient to prove that for every vertex $v \in \mathcal{S}$ at least one connecting edge cannot be intersected by a separation edge. For the sake of contradiction we assume that there exists a set E of separation edges such that every connecting edge is intersected by at least one $e \in E$.



■ **Figure 10** The decomposition of a fixed-edge graph into polygons. We have $c_1 = 4$, $c_2 = 0$, $c_3 = 1$ and $c_4 = 1$ for the number of components in each polygon. Thus, we have $c_{\max} = 4$. Note that the component inside P_4 is not counted towards c_3 , but to c_4 .



■ **Figure 11** The Delaunay Neighbourhood of a point v and a cycle of separation edges given in red.

In a first step we can prove that at least one endpoint of any $e \in E$ must be part of the Delaunay neighbourhood of v . Additionally, we can show that no boundary edge \overline{uw} can be intersected by a separation edge for \overline{vu} and a separation edge for \overline{vw} . These observations imply that we can order the edges in E , such that for all i the separation edge e_i intersects e_{i-1} and e_{i+1} , i.e., the separation edges form a cycle as depicted in Figure 11.

Next, we show that every pair of consecutive separation edges $(\overline{u_i v_i}, \overline{u_{i+1} v_{i+1}})$ must satisfy a special property, i.e., it must hold that $u_{i+1} \in C(u_i, v_i, v_{i+1})$ and $v_{i+1} \in C(u_i, v_i, u_{i+1})$. Finally, we show that this is not possible which leads to a contradiction. ◀

It is well known [28, 16] that F_1 is connected ($c_{\max} = 0$). Silveira et al. stated in [28] that for $k > 1$ the value c_{\max} can be larger than 0. But their experiments do not yield any example for which F_2 is not connected. We complement the discussion by such an example. Additionally, we show for all $k \geq 3$ there are examples with $c_{\max} \in \Omega(n)$.

► **Observation 11.**

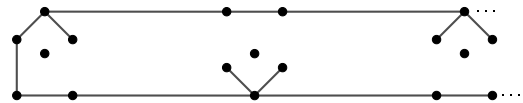
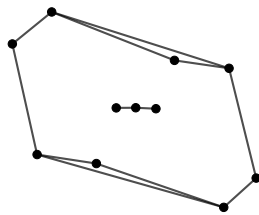
- *There exist point sets with $c_{\max} > 0$ for F_2 ; see Figure 12.*
- *For every n and $k \geq 3$ there are point sets of size n with $c_{\max} = \lfloor \frac{n}{6} \rfloor$ for F_k ; see Figure 13.*

Open question. Is there a constant d , such that F_2 has $c_{\max} \leq d$ for every point set?

Practical implications. Our results are interesting from a theoretical point of view, but the experiments in [28] with random point sets by Silveira et al. and also our own preliminary experiments indicate that for practical datasets c_{\max} is small for $k \leq 7$. Next, we confirm this assumption for the tide gauge dataset which is used for the sea surface reconstruction.

6 Experiments

We start this section by discussing the datasets. Next, we discuss the fixed-edge graphs of the tide gauge dataset. Afterwards, we provide the reconstruction process and our experimental setup. Finally, we present our results regarding the runtime and quality.



■ **Figure 12** An example with disconnected F_2 . ■ **Figure 13** An example with $c_{\max} = \frac{n}{6}$ for F_3 .

6.1 The datasets

The triangulation points for the minimum-error triangulation problem are given by the monthly tide-gauge time series from the Permanent Service for Mean Sea Level (PSMSL) [26], which is further discussed in [18]. We use the revised local reference (RLR) datasets. Furthermore, we remove some stations which do not have any values in our time-frame. This results in a dataset with 1502 stations, but not all of them record monthly. Thus, we only use between 513 and 804 different stations at once for a reconstruction.

As reference data \mathcal{R} we use the satellite altimeter datasets provided by the ESA Sea Level Climate Change Initiative (SLCCI), which are given in [13] and are further discussed in [1]. They are given as monthly gridded sea level anomalies with a spatial resolution of 0.25 degrees and are available for the timespan January 1993 to December 2015.

We assume that both datasets are given in radial coordinates. Since we focus on planar triangulations, we need to use a global map projection. We chose the Lambert azimuthal projection (LAP) which unfolds the sphere onto the plane starting at an anchor point (λ_0, ϕ_0) . For our experiments the LAP has one advantage: The projection results in significantly different distributions of the stations for sufficiently different anchor points (λ_0, ϕ_0) . This allows us to perform the fixed-edge graph experiments for a wide variety of point distributions.

It is important to note that the experiments in this paper focus on the runtime of the DP algorithm for a real world application. Thus, we only de-mean the tide gauge data as discussed in [24] and do not apply any additional corrections.

6.2 The fixed-edge graphs of the tide gauge set

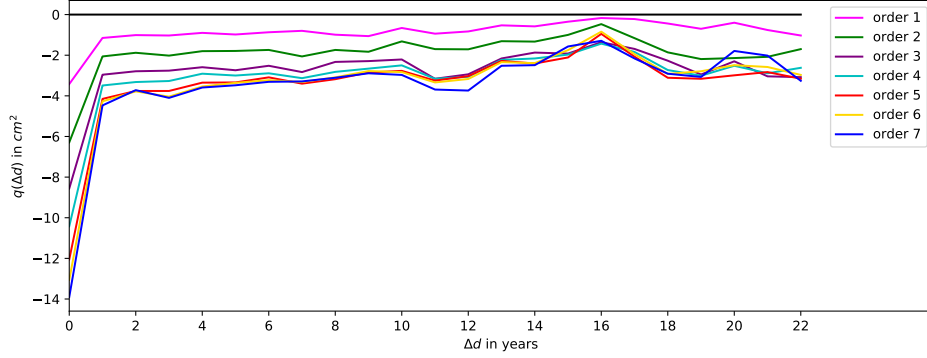
For our experiments with respect to the fixed-edge graphs we use the complete RLR dataset, i.e., all 1502 stations. We use the LAP with anchors (λ_0, ϕ_0) on an uniform 2-D 20×20 grid to generate 400 distributions of the dataset. In Table 1 the experiments are summarized. The values $\text{avg}_{c_{\max}}$ are given by the average value of c_{\max} over all samples. Additionally, we have min and max that depict the minimal and maximal value of c_{\max} for all samples. The results roughly coincide with the experiments performed on random point sets by Silveira et al. in [28] and our own preliminary experiments. The experiments suggest, that we can expect the DP algorithm to compute optimal solutions for $k \leq 7$ in reasonable time. Since Nitzke et al. suggest very small k for the reconstruction in [24], these experiments are promising.

6.3 Sea surface reconstruction

The reconstruction process can be summarized as follows: We learn a minimum-error triangulation D in some epoch i and then use it to reconstruct the sea surface at some other point in time j , by using the triangulation D with the height values of epoch j . Since not all tide gauge stations provide data for every epoch i , we need to consider the set G^{ij} which is given by all stations that have reasonable values for epoch i as well as for j . We denote the optimal triangulation using G^{ij} and the reference points A_i by D_M^{ij} . For comparison we use the Delaunay triangulation D_D^{ij} of the set G^{ij} which has already been successfully used for

■ **Table 1** The average of c_{\max} and the min/max value of c_{\max} for the projections of the RLR data.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$
$\text{avg}_{c_{\max}}$	0.00	0.00	0.45	1.20	2.05	3.68	7.11	15.88	33.16
min/max	0/0	0/0	0/2	0/3	1/5	2/12	3/18	6/38	11/82



■ **Figure 14** Averaged $q(\Delta d)$ of our approach w.r.t. the epoch difference Δd for different order k .

the sea surface reconstruction task in [25]. If we have altimeter data available for epoch j , we can evaluate the quality of our approximation. Overall the reconstruction for epoch j using i and order k can be performed as follows:

1. Compute the set G^{ij} and the k -OD triangles \mathfrak{T}^{ij} as described in [28].
2. Compute the weights $w_T(A_i)$ of all $T \in \mathfrak{T}^{ij}$ with respect to A_i as discussed in Section 2.
3. Compute the optimal k -OD triangulation D_M^{ij} with the DP algorithm given in Section 5 and also compute the Delaunay triangulation D_D^{ij} .
4. Evaluate the quality of the triangulations with respect to A_j .

For the evaluation we compute the *empirical variance* of a triangulation

$$\sigma_{ij}^2(D) = \frac{1}{n-1} \sum_{T \in D} \sum_{a \in A_j, a \in T} (s_T(a) - h_j(a))^2,$$

where n is the number of altimeter points in $\text{conv}(D)$. Note that this is exactly the average minimum error. Additionally, we define the *variance reduction* of a reconstruction by

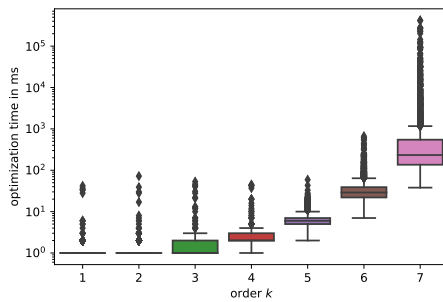
$$\Delta\sigma_{ij}^2 = \sigma_{ij}^2(D_M^{ij}) - \sigma_{ij}^2(D_D^{ij}).$$

Next, we can group together reconstructions for epochs i, j and i', j' where $|i-j| = |i'-j'|$. This allows us to define the *average variance reduction* of a temporal difference Δd by

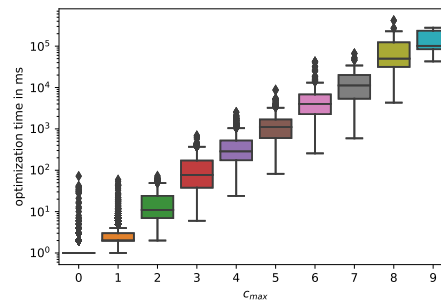
$$q(\Delta d) = \frac{1}{|\mathcal{D}(\Delta d)|} \sum_{(i,j) \in \mathcal{D}(\Delta d)} \Delta\sigma_{ij}^2.$$

The set $\mathcal{D}(\Delta d)$ is given by all tuples (i, j) with $|i-j| = \Delta d$. Using the temporal difference, we can investigate how far back in time our optimized triangulation outperforms the Delaunay triangulation (DT). Nitzke et al.[24] noticed that q has a seasonal behaviour, i.e., q has local maxima every 12 month. Thus, we only use datasets with $j = i \pm 12l$ with $l \in \mathbb{N}$ for the reconstruction. A more in depth discussion of the evaluation methods can be found in [24].

Reconstruction quality. For all of the experiments we choose an LAP anchored in the Atlantic Ocean, namely $(-40, 16)$. We compute all possible reconstructions for epochs i and j with $i \geq j$ for the orders $k \leq 7$, i.e., we use every epoch i for training and validate the learned triangulation on all possible epochs j with $j = i - 12l$. Next, we group them with respect to Δd . In Figure 14 the $q(\Delta d)$ values are depicted. Recall that our approach performs better than the DT, if $q(\Delta d) < 0$. It should be mentioned, however, that for $\Delta d \geq 18$ the quality of the experiments deteriorates, since only few samples span this epoch difference.



■ **Figure 15** Optimization time depending on the order.



■ **Figure 16** Optimization time depending on c_{\max} .

Note that the variance reductions for $\Delta d = 0$ are far better than for larger Δd , since the reconstruction epoch is the same as the training epoch. The variance reductions for order 1 and order 2 are smoother, but also worse than the ones for higher orders. For $\Delta d > 10$ the variance reductions for the orders 3–6 are very similar and even order 7 is comparable. The aforementioned orders also share local extrema at $\Delta d = 10, 11, 18, 20$. For order 7 the extrema become more pronounced which leads to better minima but also to worse maxima. Note that calculating the empirical variances $\sigma_{ij}^2(D_D^{ij})$ for all epochs yields values between 80cm^2 and 120cm^2 . Hence, for example, an absolute variance reduction of 2cm^2 roughly coincides with a relative variance reduction of 2%.

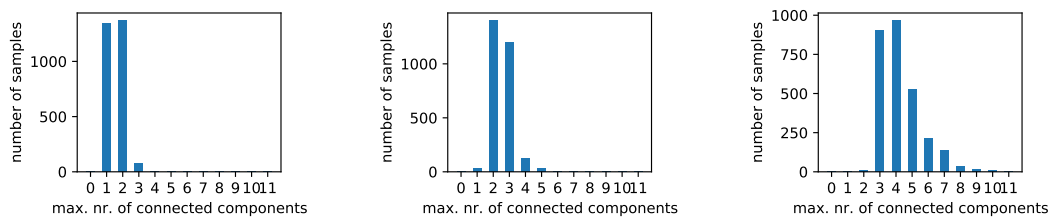
The overall variance reduction gets better for higher orders. This is contrary to the results by Nitzke et al. [24], who suggested $k = 1, 2$ for the reconstruction. This difference may have geometric reasons, i.e., the points in the North Sea dataset used in [24] more or less trace a polygon without inner points and our global datasets have a more arbitrary distribution. Moreover, the LAP distorts distances as well as angles which may also contribute to the different results for the local and global datasets.

Runtime. For the experiments we used a machine with an *AMD Ryzen 5 3600 6-Core Processor* clocked at *4.4 GHz* and *16 GB RAM*. We did not implement the geometric pre-processing as discussed in [16]. Our pre-processing has roughly cubic runtime (3–4 seconds per reconstruction). For larger orders k we expect the optimization to dominate the runtime.

The optimization time with respect to the order is given in Figure 15. Note that the optimization time for $k \leq 5$ is at most 30ms. For $k = 6$ the average runtime is still low with roughly 50ms. For $k = 7$ most datasets can be optimized in a few seconds, but some need around 20 minutes for the optimization and five datasets reach a cut-off time of one hour.

The box-plot in Figure 16 depicts the runtime with respect to the number of connected components c_{\max} . The logarithmic scaling nicely illustrates the exponential increase. If we also consider the distribution of c_{\max} for the different datasets and orders, we can easily connect the two box-plots. For $k \leq 4$ all of the datasets have $c_{\max} \leq 2$. Thus, the maximal runtime for orders $k \leq 4$ matches the worst runtime for $c_{\max} \leq 2$. For orders $k = 5, 6, 7$ the c_{\max} distributions are illustrated in Figure 17. Note that for $k = 5$ and $k = 6$ most datasets still have $c_{\max} \leq 2$ which results in the very low average runtime. For $k = 7$ the distribution starts to shift towards higher c_{\max} which results in the higher average runtime.

In summary, our experiments show that for our datasets we can compute k -OD min-error triangulations for $k \leq 6$ and also for $k = 7$ except for a few samples in reasonable time.



■ **Figure 17** The c_{\max} distribution of the reconstruction datasets for orders $k = 5, 6, 7$.

7 Conclusion

We prove that it is NP-hard to approximate an optimal solution to the minimum-error triangulation problem. Our results also imply the inapproximability of the following generalization: minimizing the distance between s_D and h on \mathcal{R} for any metric on \mathbb{R}^m , especially the L_p -metric $(\sum_{r \in \mathcal{R}} |s_D(r) - h(r)|^p)^{1/p}$ for $p \in [1, \infty)$ and the L_∞ -metric $\max_{r \in \mathcal{R}} |s_D(r) - h(r)|$. Additionally, we apply the dynamic programming algorithm by Silveira et al. [28] to minimum-error triangulations and extend their experiments, regarding the fixed edges to a real world dataset. We further investigate the fixed-edge graphs for order $k = 2$ and give a worst-case example for $k = 3$. Finally, we perform the dynamic sea surface reconstruction similar to Nitzke et al. in [24] for significantly larger datasets using a new algorithmic approach.

A future line of research is the extension of the dynamic programming algorithm to datasets on the sphere, i.e., spherical triangulations. This would allow a more realistic reconstruction of the global dynamic sea surface. A combination with ILP techniques will be a further step [14]. It would also be interesting to include multiple datasets for the learning of the reconstruction triangulation. We believe that our work will open the door for the application of optimal triangulation approaches to the problem of multi-decadal global sea level reconstructions from tide gauge data. In addition, with the growing amount of satellite and in-situ ocean sensors (buoys, Argo floats, ...) we see potential for a more widespread application of triangulation methods in generating gridded ocean data products.

References

- 1 M. Ablain, A. Cazenave, G. Larnicol, M. Balmaseda, P. Cipollini, Y. Faugère, M. J. Fernandes, O. Henry, J. A. Johannessen, P. Knudsen, O. Andersen, J. Legeais, B. Meyssignac, N. Picot, M. Roca, S. Rudenko, M. G. Scharffenberg, D. Stammer, G. Timms, and J. Benveniste. Improved sea level record over the satellite altimetry era (1993–2010) from the Climate Change Initiative project. *Ocean Science*, 11(1):67–82, 2015. doi:10.5194/os-11-67-2015.
- 2 Pankaj K. Agarwal and Subhash Suri. Surface approximation and geometric partitions. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 24–33, USA, 1994. Society for Industrial and Applied Mathematics.
- 3 Lyuba Alboul, Gertjan Kloosterman, Cornelis Traas, and Ruud van Damme. Best data-dependent triangulations. *Journal of Computational and Applied Mathematics*, 119(1):1–12, 2000. doi:10.1016/S0377-0427(00)00368-X.
- 4 Efthymios Anagnostou and Derek Corneil. Polynomial-time instances of the minimum weight triangulation problem. *Computational Geometry*, 3(5):247–259, 1993. doi:10.1016/0925-7721(93)90016-Y.
- 5 Anna Arutyunova, Anne Driemel, Jan-Henrik Haunert, Herman Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel, and Heiko Röglin. Minimum-error triangulations for sea surface reconstruction. 2022. arXiv:2203.07325v1.

- 6 Marshall Bern and David Eppstein. Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry*, 1992. doi:10.1142/9789814355858_0002.
- 7 Magdalene Borgelt, Christian Borgelt, and Christos Levcopoulos. Fixed parameter algorithms for the minimum weight triangulation problem. *Int. J. Comput. Geometry Appl.*, 18:185–220, June 2008. doi:10.1142/S0218195908002581.
- 8 Jeffrey L. Brown. Vertex based data dependent triangulations. *Computer Aided Geometric Design*, 8(3):239–251, 1991. doi:10.1016/0167-8396(91)90008-Y.
- 9 Siu-Wing Cheng, Mordecai J. Golin, and Jeffrey Tsang. Expected case analysis of {221}-skeletons with applications to the construction of minimum-weight triangulations. Master's thesis, Hong Kong University of Science and Technology, 1995.
- 10 John A. Church, Neil J. White, Richard Coleman, Kurt Lambeck, and Jerry X. Mitrovica. Estimates of the Regional Distribution of Sea Level Rise over the 1950–2000 Period. *Journal of Climate*, 17(13):2609–2625, July 2004. doi:10.1175/1520-0442(2004)017<2609:EOTRDO>2.0.CO;2.
- 11 Thierry de Kok, Marc van Kreveld, and Maarten Löffler. Generating realistic terrains with higher-order Delaunay triangulations. *Computational Geometry*, 36(1):52–65, 2007. Special Issue on the 21st European Workshop on Computational Geometry. doi:10.1016/j.comgeo.2005.09.005.
- 12 Nira Dyn, David Levin, and Samuel Rippa. Data Dependent Triangulations for Piecewise Linear Interpolation. *IMA Journal of Numerical Analysis*, 10(1):137–154, January 1990. doi:10.1093/imanum/10.1.137.
- 13 ESA. Sea level CCI ECV dataset: Time series of gridded sea level anomalies(sla), 2021. European Space Agency (ESA). URL: <https://catalogue.ceda.ac.uk/uuid/142052b9dc754f6da47a631e35ec4609>.
- 14 S.P. Fekete, A. Haas, Y. Lieder, E. Niehs, M. Perk, V. Sack, and C. Scheffer. On hard instances of the minimum-weight triangulation problem. In *36th European Workshop on Computational Geometry (EuroCG 2020)*, March 2020.
- 15 P. Gilbert. New results on planar triangulations. Master's thesis, University of Illinois, Coordinated Science Lab, Urbana, IL, USA, 1979.
- 16 Joachim Gudmundsson, Mikael Hammar, and Marc van Kreveld. Higher order Delaunay triangulations. *Computational Geometry*, 23(1):85–98, 2002. doi:10.1016/S0925-7721(01)00027-X.
- 17 Andreas Haas. Solving large-scale minimum-weight triangulation instances to provable optimality. In Bettina Speckmann and Csaba D. Tóth, editors, *34th International Symposium on Computational Geometry, SoCG 2018*, volume 99 of *LIPICs*, pages 44:1–44:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.SoCG.2018.44.
- 18 Simon J. Holgate, Andrew Matthews, Philip L. Woodworth, Lesley J. Rickards, Mark E. Tamisiea, Elizabeth Bradshaw, Peter R. Foden, Kathleen M. Gordon, Svetlana Jevrejeva, and Jeff Pugh. New Data Systems and Products at the Permanent Service for Mean Sea Level. *Journal of Coastal Research*, 29(3):493–504, December 2012. doi:10.2112/JCOASTRES-D-12-00175.1.
- 19 G.T. Klincsek. Minimal triangulations of polygonal domains. In Peter L. Hammer, editor, *Combinatorics 79*, volume 9 of *Annals of Discrete Mathematics*, pages 121–123. Elsevier, 1980. doi:10.1016/S0167-5060(08)70044-X.
- 20 Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427, 1992.
- 21 Charles L. Lawson. Software for C^1 surface interpolation. In John R. Rice, editor, *Mathematical Software*, pages 161–194. Academic Press, 1977. doi:10.1016/B978-0-12-587260-7.50011-X.
- 22 David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11:329–343, 1982.
- 23 Wolfgang Mulzer and Günter Rote. Minimum-weight triangulation is NP-hard. *Journal of the ACM*, 55(2):1–29, May 2008. doi:10.1145/1346330.1346336.

7:18 Minimum-Error Triangulations for Sea Surface Reconstruction

- 24 Alina Nitzke, Benjamin Niedermann, Luciana Fenoglio-Marc, Jürgen Kusche, and Jan-Henrik Haunert. Reconstructing the dynamic sea surface from tide gauge records using optimal data-dependent triangulations. *Computers & Geosciences*, 157:104920, 2021. doi:10.1016/j.cageo.2021.104920.
- 25 Marco Olivieri and Giorgio Spada. Spatial sea-level reconstruction in the Baltic Sea and in the Pacific Ocean from tide gauges observations. *Annals of Geophysics*, 59(3), 2016. doi:10.4401/ag-6966.
- 26 Permanent service for mean sea level (PSMSL), 2021. Retrieved 19 Apr 2021 from <http://www.psmsl.org/data/obtaining/>.
- 27 Natalia Rodríguez and Rodrigo I. Silveira. Implementing data-dependent triangulations with higher order Delaunay triangulations. *ISPRS International Journal of Geo-Information*, 6(12), 2017. doi:10.3390/ijgi6120390.
- 28 Rodrigo I. Silveira and Marc van Kreveld. Optimal higher order Delaunay triangulations of polygons. *Computational Geometry*, 42(8):803–813, 2009. Special Issue on the 23rd European Workshop on Computational Geometry. doi:10.1016/j.comgeo.2008.02.006.
- 29 Kai Wang, Chor-Pang Lo, George A. Brook, and Hamid R. Arabnia. Comparison of existing triangulation methods for regularly and irregularly spaced height fields. *International Journal of Geographical Information Science*, 15(8):743–762, 2001. doi:10.1080/13658810110074492.

Delaunay-Like Triangulation of Smooth Orientable Submanifolds by ℓ_1 -Norm Minimization

Dominique Attali ✉

Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, Grenoble, France

André Lieutier ✉

Dassault systèmes, Aix-en-Provence, France

Abstract

In this paper, we focus on one particular instance of the shape reconstruction problem, in which the shape we wish to reconstruct is an orientable smooth submanifold of the Euclidean space. Assuming we have as input a simplicial complex K that approximates the submanifold (such as the Čech complex or the Rips complex), we recast the reconstruction problem as a ℓ_1 -norm minimization problem in which the optimization variable is a chain of K . Providing that K satisfies certain reasonable conditions, we prove that the considered minimization problem has a unique solution which triangulates the submanifold and coincides with the flat Delaunay complex introduced and studied in a companion paper [3]. Since the objective is a weighted ℓ_1 -norm and the constraints are linear, the triangulation process can thus be implemented by linear programming.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases manifold reconstruction, Delaunay complex, triangulation, sampling conditions, optimization, ℓ_1 -norm minimization, simplicial complex, chain, fundamental class

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.8

Related Version *Full Version*: <https://arxiv.org/abs/2203.06008>

Acknowledgements We are grateful to the anonymous referees for carefully reading the paper and many helpful suggestions.

1 Introduction

In many practical situations, the shape of interest is only known through a finite set of data points. Given these data points as input, it is then natural to try to construct a *triangulation* of the shape, that is, a set of simplices whose union is homeomorphic to the shape. This paper focuses on one particular instance of this problem, in which the shape we wish to reconstruct is a smooth d -dimensional submanifold of the Euclidean space. We show that, when the submanifold is orientable and under appropriate conditions, a triangulation of that submanifold can be expressed as the solution of a weighted ℓ_1 -norm minimization problem under linear constraints. This formulation gives rise to new algorithms for the triangulation of manifolds, in particular when the manifolds have large codimensions.

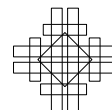
Variational formulation of Delaunay triangulation and generalizations. Our work is based on the observation that when we consider a point cloud P in \mathbb{R}^d , its Delaunay complex can be expressed as the solution of a particular ℓ_p -norm minimization problem. This fact is best explained by lifting the point set P vertically onto the paraboloid $\mathcal{P} \subseteq \mathbb{R}^{d+1}$ whose equation is $x_{d+1} = \sum_{i=1}^d x_i^2$. It is well-known that the Delaunay complex of P is isomorphic to the boundary complex of the lower convex hull of the lifted points \hat{P} .

Starting from this equivalence, Chen has observed in [16] that the Delaunay complex of P minimizes the ℓ_p -norm of the difference between two functions over all triangulations T of P . The graph of the first function is the lifted triangulation \hat{T} and the graph of the second one is



© Dominique Attali and André Lieutier;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 8; pp. 8:1–8:16
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



the paraboloid \mathcal{P} . This variational formulation has been successfully exploited in [1, 14, 17] for the generation of *Optimal Delaunay Triangulations*. When $p = 1$, the ℓ_p -norm associated to T is what we call in this paper the *Delaunay energy* of T and, can be interpreted as the volume enclosed between the lifted triangulation \hat{T} and the paraboloid \mathcal{P} .

Our contribution. While it seems difficult to extend the lifting construction when points of P sample a d -dimensional submanifold of \mathbb{R}^N , our main result is to show that nonetheless, the induced variational formulation can still be transposed.

Consider a set of points P that sample a d -dimensional submanifold \mathcal{M} . When searching for a triangulation of \mathcal{M} from P , it seems reasonable to restrict the search within a simplicial complex K built from P . A first crucial ingredient in our work is to embed the triangulations that one can build using simplices of K inside the vector space formed by simplicial d -cycles¹ of K over the field \mathbb{R} . In spirit, this is similar to what is done in the theory of minimal surfaces, when oriented surfaces are considered as particular elements of a much larger set, namely the space of currents [25], that enjoys the nice property of being a vector space. Going back to the case of points in the Euclidean space, if one minimizes the Delaunay energy in the larger set of simplicial chains with real coefficients and under adequate boundary constraints, one obtains a particular chain with coefficients in $\{0, 1\}$ whose simplices, roughly speaking, do not “overlap”. The support of that chain, that is the set of simplices with coefficient 1, coincides with the Delaunay triangulation. The proof is quite direct and relies on the geometric interpretation provided by the lifting construction [18, 31].

We show that when transposing this to the case of points P that sample a d -dimensional submanifold \mathcal{M} , minimizing the Delaunay energy provides indeed a triangulation of \mathcal{M} . The proof requires us to introduce a more elaborate construction, the *Delloc complex* of P , as a tool to describe the solution. The d -simplices of that complex possess exactly the property that we need for our analysis. In a companion paper [3] we show that the Delloc complex indeed provides a triangulation of the manifold, assuming the set of sample points P to be sufficiently dense, safe, and not too noisy. Incidentally, the Delloc complex coincides with the *flat Delaunay complex* introduced in our companion paper [3] and is akin to the *tangential Delaunay complex* introduced and studied in [5, 6]. When the manifold is sufficiently densely sampled by the data points, all three constructions are locally isomorphic to a (weighted) Delaunay triangulation computed in a local tangent space to the manifold. Intuitively, this indicates that the Delaunay energy should locally reach a minimum for all three constructions and, therefore ought to be also a global minimum. Actually, turning this intuitive reasoning into a correct proof turns out to be more tricky than it appears and is the main purpose of the present paper. In particular, we need to globally compare the Delaunay energy of the cycle carrying the Delloc complex with that of alternate d -cycles, and this requires us to carefully distribute the Delaunay energy along barycentric coordinates (see Section 6).

Algorithms. Several authors, with computational topology or topological data analysis motivations, have considered the computation of ℓ_1 -minimum homology representative cycles, [13, 9, 19, 10, 20], generally for integers or integers modulo p coefficients. The celebrated sparsity of ℓ_1 -minima manifests itself in this context by the fact that the support of such minima is sparse, in other words it is non-zero only on a small subset of simplices of K .

Note that an alternative algorithm to the one proposed in this paper could be to compute a triangulation of \mathcal{M} by returning such a minimal sparse representative. Indeed, when the data points sample sufficiently densely and accurately the manifold compared to the reach of the

¹ Or relative d -cycles when the considered domain has a boundary.

manifold, one could – in theory – take either the Čech complex or the Vietoris-Rips complex as the complex K , since it is known that by choosing the scale parameter of these complexes carefully, they are guaranteed to have the same homotopy type as \mathcal{M} [12, 11, 4, 29, 24]. Recall that, when \mathcal{M} is orientable and connected, its d -homology group with real coefficients is one-dimensional, and a normalized generator of it is called the *manifold fundamental class*. Hence, when K and \mathcal{M} are homotopy equivalent, the d -homology group of K is also one-dimensional. It follows that extracting any non-boundary cycle of K (using standard linear algebra operations on the boundary operators ∂_d and ∂_{d+1} of K) provides a d -cycle γ_0 which is, up to a multiplicative constant, a representative of a generator of the fundamental class of \mathcal{M} . An alternate algorithm could then search, among chains homologous to γ_0 , for the one with the minimal Delaunay energy. The solution of the corresponding linear optimization problem would then be a chain which carries the Delloc complex. While elegant in theory, the size required for the $(d+1)$ -skeleton of the Čech or Vietoris-Rips complex may be prohibitive in practice.

Instead, we describe a procedure that only requires the milder condition on K to be a simplicial complex large enough to contain the Delloc complex, at the cost of adding a certain form of normalization constraint. For the purpose of the proof, it is convenient to first consider a rather artificial problem, where, besides the sample P , the manifold \mathcal{M} is known. In the full version [2], we show how to turn this problem into a more realistic one that takes as input only the sample of the unknown manifold, and is correct assuming that reasonable sampling conditions hold. While we do not yet explore practical efficient algorithms in this paper, the minimization of a ℓ_1 -norm under linear constraints in \mathbb{R}^n , where n is the number of d -simplices in the considered simplicial complex K , can be turned into a linear optimization problem in the standard form through slack variables, and can be addressed by standard linear programming techniques such as the simplex algorithm.

The missing proofs may be found in the full version [2].

2 Preliminaries

In this section, we review the necessary background and explain some of our terms.

2.1 Subsets and submanifolds

Given a subset $A \subseteq \mathbb{R}^N$, the affine space spanned by A is denoted by $\text{aff } A$ and the convex hull of A by $\text{conv } A$. The *medial axis* of A , denoted as $\text{axis}(A)$, is the set of points in \mathbb{R}^N that have at least two closest points in A . By definition, the *projection map* $\pi_A : \mathbb{R}^N \setminus \text{axis}(A) \rightarrow A$ associates to each point x its unique closest point in A . The *reach* of A is the infimum of distances between A and its medial axis, and is denoted as $\text{reach } A$. By definition, the projection map π_A is well-defined on every subset of \mathbb{R}^N that does not intersect the medial axis of A . In particular, recalling that the *r -tubular neighborhood* of A is the set of points $A^{\oplus r} = \{x \in \mathbb{R}^N \mid d(x, A) \leq r\}$, the projection map π_A is well-defined on every r -tubular neighborhood of A with $r < \text{reach } A$. We denote the ball centered at $x \in \mathbb{R}^N$ and with radius $\rho \in \mathbb{R}$ by $B(x, \rho)$. For short, we say that a subset $\sigma \subseteq \mathbb{R}^N$ is ρ -small if it can be enclosed in a ball of radius ρ .

Throughout the paper, \mathcal{M} designates a compact connected orientable C^2 d -dimensional submanifold of \mathbb{R}^N for $d < N$. For any point $m \in \mathcal{M}$, the tangent plane to m at \mathcal{M} is denoted as $\mathbf{T}_m \mathcal{M}$. Because \mathcal{M} is C^2 and therefore $C^{1,1}$, the reach of \mathcal{M} is positive [23]. We let \mathcal{R} be a fixed finite constant such that $0 < \mathcal{R} \leq \text{reach } \mathcal{M}$.

2.2 Simplicial complexes

In this section, we review some background notation on simplicial complexes [27]. We also introduce the concept of faithful reconstruction which encapsulates what we mean by a “desirable” approximation of a manifold.

All simplices and simplicial complexes that we consider in the paper are abstract. Each abstract simplex $\sigma \subseteq \mathbb{R}^N$ is naturally associated to a geometric simplex defined as $\text{conv } \sigma$. The dimension of $\text{conv } \sigma$ is the dimension of the affine space $\text{aff } \sigma$, and cannot be larger than the dimension of the abstract simplex σ . When the dimension of the geometric simplex $\text{conv } \sigma$ coincides with that of the abstract simplex σ , we say that σ is *non-degenerate*. Equivalently, the vertices of σ form an affinely independent set of points. The *star* of $x \in \mathbb{R}^N$ in a simplicial complex K is $\text{St}(x, K) = \{\sigma \in K \mid x \in \text{conv } \sigma\}$.

Given a set of simplices Σ with vertices in \mathbb{R}^N (not necessarily forming a simplicial complex), we let $\Sigma^{[d]}$ designate the d -simplices of Σ . We define the *shadow* of Σ as the subset of \mathbb{R}^N covered by the relative interior of the geometric simplices associated to the abstract simplices in Σ , $|\Sigma| = \bigcup_{\sigma \in \Sigma} \text{relint}(\text{conv } \sigma)$. We shall say that Σ is *geometrically realized* (or *embedded*) if (1) $\dim(\sigma) = \dim(\text{aff } \sigma)$ for all $\sigma \in \Sigma$, and (2) $\text{conv}(\alpha \cap \beta) = \text{conv } \alpha \cap \text{conv } \beta$ for all $\alpha, \beta \in \Sigma$.

► **Definition 1 (Faithful reconstruction).** *Consider a subset $A \subseteq \mathbb{R}^N$ whose reach is positive, and a simplicial complex K with a vertex set in \mathbb{R}^N . We say that K reconstructs A faithfully (or is a faithful reconstruction of A) if the following three conditions hold:*

Embedding: K is geometrically realized;

Closeness: $|K|$ is contained in the r -tubular neighborhood of A for some $0 \leq r < \text{reach } A$;

Homeomorphism: The restriction of $\pi_A : \mathbb{R}^N \setminus \text{axis}(A) \rightarrow A$ to $|K|$ is a homeomorphism.

2.3 Height, circumsphere and smallest enclosing ball

The *height* of a simplex σ is $\text{height}(\sigma) = \min_{v \in \sigma} d(v, \text{aff}(\sigma \setminus \{v\}))$. The height of σ vanishes if and only if σ is degenerate. If σ is non-degenerate, then, letting $d = \dim \sigma = \dim \text{aff } \sigma$, there exists a unique $(d - 1)$ -sphere that circumscribes σ and therefore at least one $(N - 1)$ -sphere that circumscribes σ . Hence, if σ is non-degenerate, it makes sense to define $S(\sigma)$ as the smallest $(N - 1)$ -sphere that circumscribes σ . Let $Z(\sigma)$ and $R(\sigma)$ denote the center and radius of $S(\sigma)$, respectively. Let c_σ and r_σ denote the center and radius of the smallest N -ball enclosing σ , respectively. Clearly, $r_\sigma \leq R(\sigma)$ and both c_σ and $Z(\sigma)$ belong to $\text{aff } \sigma$. The intersection $S(\sigma) \cap \text{aff } \sigma$ is a $(d - 1)$ -sphere which is the unique $(d - 1)$ -sphere circumscribing σ in $\text{aff } \sigma$.

2.4 Delaunay complexes

Consider a finite point set $Q \subseteq \mathbb{R}^N$. We say that $\sigma \subseteq Q$ is a *Delaunay simplex* of Q if there exists an $(N - 1)$ -sphere S that circumscribes σ and such that no points of Q belong to the open ball whose boundary is S . The set of Delaunay simplices form a simplicial complex called the *Delaunay complex* of Q and denoted as $\text{Del}(Q)$.

► **Definition 2 (General position).** *Let $d = \dim(\text{aff } Q)$. We say that $Q \subseteq \mathbb{R}^N$ is in general position if no $d + 2$ points of Q lie on a common $(d - 1)$ -dimensional sphere.*

► **Lemma 3.** *When Q is in general position, $\text{Del}(Q)$ is geometrically realized.*

Let us recall a famous result which says that building a Delaunay complex in \mathbb{R}^N is topologically equivalent to building a lower convex hull in \mathbb{R}^{N+1} . For simplicity, we shall identify each point $x \in \mathbb{R}^N$ with a point $(x, 0)$ in \mathbb{R}^{N+1} . Consider the paraboloid $\mathcal{P} \subseteq \mathbb{R}^{N+1}$

defined as the graph of the function $\|\cdot\|^2 : \mathbb{R}^N \rightarrow \mathbb{R}, x \mapsto \|x\|^2$, where $\|\cdot\|$ designates the Euclidean norm. For each point $x \in \mathbb{R}^N$, its vertical projection onto \mathcal{P} is the point $\hat{x} = (x, \|x\|^2) \in \mathbb{R}^{N+1}$, which we call the *lifted image* of x . Similarly, the lifted image of $Q \subseteq \mathbb{R}^N$ is $\hat{Q} = \{\hat{q} \mid q \in Q\}$. Recall that the lower convex hull of \hat{Q} is the portion of $\text{conv } \hat{Q}$ visible to a viewer standing at $x_{d+1} = -\infty$. A classical result says that σ is a Delaunay simplex of Q if and only if $\text{conv } \hat{\sigma}$ is contained in the lower convex hull of \hat{Q} [22].

2.5 Delaunay energy for triangulations

We recall that a *triangulation* T of Q designates a simplicial complex with vertex set Q which is geometrically realized and whose shadow covers $\text{conv } Q$. It is well-known that the Delaunay complex of Q optimizes many functionals over the set of triangulations of Q [8, 30, 28], one of them being the Delaunay energy that we shall now define [15]. Let $d = \dim(\text{aff } Q)$. Given a triangulation T of Q , the *Delaunay energy* $E_{\text{del}}(T)$ of T is defined as the $(d+1)$ -volume between the d -manifold $|\hat{T}| = \bigcup_{\sigma \in T} \text{conv } \hat{\sigma}$ and the paraboloid \mathcal{P} . Let us derive an expression for this $(d+1)$ -volume. Consider a point $x \in \text{conv } Q$. By construction, x belongs to at least one geometric d -simplex $\text{conv } \sigma$ for some $\sigma \in T$. Erect an infinite vertical half-line going up from x . This half-line intersects the paraboloid \mathcal{P} at point \hat{x} and the lifted geometric d -simplex $\text{conv } \hat{\sigma}$ at point x_σ^* . We have

$$E_{\text{del}}(T) = \sum_{\sigma} \int_{x \in \text{conv } \sigma} \|\hat{x} - x_\sigma^*\| dx.$$

► **Theorem 4** (Delaunay complex by a variational approach). *When Q is in general position, the triangulation of Q that minimizes the Delaunay energy is unique and equals $\text{Del}(Q)$.*

Theorem 4 is a direct consequence of the lifting construction [28, 16].

2.6 Delaunay weight

To each non-degenerate d -dimensional abstract simplex $\alpha \in \mathbb{R}^N$ we assign a non-negative real number that we call the Delaunay weight of α . The reasons for this will become clear shortly. Let $\alpha \subseteq \mathbb{R}^N$ be a non-degenerate abstract simplex. We recall that the power distance of point $x \in \mathbb{R}^N$ from $S(\alpha)$ is $\text{Power}_\alpha(x) = \|x - Z(\alpha)\|^2 - R(\alpha)^2$.

► **Definition 5** (Delaunay weight). *The Delaunay weight of a non-degenerate simplex α is:*

$$\omega(\alpha) = - \int_{x \in \text{conv } \alpha} \text{Power}_\alpha(x) dx.$$

Easy computations show that $\text{Power}_\alpha(x) = -\|\hat{x} - x_\alpha^*\|$; see for instance [21]. Hence, if $d = \dim(\alpha)$, we see that $\omega(\alpha)$ represents the $(d+1)$ -volume between the lifted geometric simplex $\text{conv } \hat{\alpha}$ and the paraboloid \mathcal{P} . Therefore the Delaunay energy can be expressed as $E_{\text{del}}(T) = \sum_{\alpha} \omega(\alpha)$, where α ranges over all d -simplices of T . Below, we give a closed expression for the Delaunay weight due to Chen and Holst in [14]. Writing $\text{vol}(\alpha)$ for the d -dimensional volume of $\text{conv } \alpha$, we have:

► **Lemma 6** ([14]). *The weight of the non-degenerate d -simplex $\alpha = \{a_0, \dots, a_d\}$ is*

$$\omega(\alpha) = \frac{1}{(d+1)(d+2)} \text{vol}(\alpha) \left[\sum_{0 \leq i < j \leq d} \|a_i - a_j\|^2 \right].$$

The expression of the Delaunay weight given in Lemma 6 shows that two simplices that are isometric have the same Delaunay weight. Hence, a Delaunay energy can be straightforwardly associated to any “soup” Σ of d -simplices living in \mathbb{R}^N by setting $E(\Sigma) = \sum_{\sigma \in \Sigma} \omega(\sigma)$. It is then tempting to ask what would happen if one minimizes this energy when the vertices of Σ sample a d -manifold.

2.7 Chains and weighted norms

In this section, we recall some standard notation concerning chains. Chains play an important role in this work as they provide a tool to embed the discrete set of candidate solutions (faithful reconstructions of \mathcal{M}) into a larger continuous space. Consider an abstract simplicial complex K and assume that each simplex σ in K is given an arbitrary orientation. A d -chain of K with coefficients in \mathbb{R} is a formal sum $\gamma = \sum_{\sigma} \gamma(\sigma)\sigma$, where σ ranges over all d -simplices of K and $\gamma(\sigma) \in \mathbb{R}$ is the value (or the coordinate) assigned to the oriented d -simplex σ . The set of such d -chains is a vector space denoted by $C_d(K, \mathbb{R})$. Recall that the ℓ_1 -norm of γ is defined by $\|\gamma\|_1 = \sum_{\sigma} |\gamma(\sigma)|$. Let W be a weight function which assigns a non-negative weight $W(\sigma)$ to each d -simplex σ of K . The W -weighted ℓ_1 -norm of γ is expressed as $\|\gamma\|_{1,W} = \sum_{\sigma} W(\sigma)|\gamma(\sigma)|$. We shall say that a chain γ is *carried by* a subcomplex D of K if γ has value 0 on every simplex that is not in D . The *support* of γ is the set of simplices on which γ has a non-zero value. It is denoted by $\text{Supp } \gamma$.

3 Delloc complex

Given a finite set of points P in \mathbb{R}^N , a dimension d , and a scale parameter ρ , we introduce a construction which we call the d -dimensional Delloc complex of P at scale ρ . First, we define the property for a simplex to be delloc.

► **Definition 7** (Delloc complex). *We say that a simplex σ is delloc in P at scale ρ if $\sigma \in \text{Del}(\pi_{\text{aff } \sigma}(P \cap B(c_{\sigma}, \rho)))$. The d -dimensional Delloc complex of P at scale ρ is the set of d -simplices that are delloc in P at scale ρ together with all their faces, and is denoted by $\text{Delloc}_d(P, \rho)$.*

We now state a theorem which establishes conditions under which the Delloc complex is a faithful reconstruction of \mathcal{M} . The theorem can be seen as a corollary of the main theorem that we establish in the companion paper [3]. We need some notations and definitions.

► **Definition 8** (Dense, accurate, and separated). *We say that P is an ε -dense sample of \mathcal{M} if for every point $m \in \mathcal{M}$, there is a point $p \in P$ with $\|p - m\| \leq \varepsilon$ or, equivalently, if $\mathcal{M} \subseteq P^{\oplus \varepsilon}$. We say that P is a δ -accurate sample of \mathcal{M} if for every point $p \in P$, there is a point $m \in \mathcal{M}$ with $\|p - m\| \leq \delta$ or, equivalently, if $P \subseteq \mathcal{M}^{\oplus \delta}$. Let $\text{separation}(P) = \min_{p \neq q \in P} \|p - q\|$.*

We stress that our definition of a protected simplex differs slightly from the one in [7, 6].

► **Definition 9** (Protection). *We say that a non-degenerate simplex $\sigma \subseteq \mathbb{R}^N$ is ζ -protected with respect to $Q \subseteq \mathbb{R}^N$ if for all $q \in Q \setminus \sigma$, we have $d(q, S(\sigma)) > \zeta$.*

Let $\mathcal{H}(\sigma) = \{\mathbf{T}_m \mathcal{M} \mid m \in \pi_{\mathcal{M}}(\text{conv } \sigma)\} \cup \{\text{aff } \sigma\}$, and $\Theta(\sigma) = \max_{H_0, H_1 \in \mathcal{H}(\sigma)} \angle(H_0, H_1)$.

To the pair (P, ρ) we now associate three quantities that describe the quality of P at scale ρ :

- $\text{height}(P, \rho) = \min_{\sigma} \text{height}(\sigma)$, where the minimum is over all ρ -small d -simplices $\sigma \subseteq P$;
- $\Theta(P, \rho) = \max_{\sigma} \Theta(\sigma)$, where σ ranges over all ρ -small d -simplices of P ;
- $\text{protection}(P, \rho) = \min_{\sigma} \min_q d(q, S(\sigma))$, where the minima are over all ρ -small d -simplices $\sigma \subseteq P$ and all points $q \in \pi_{\text{aff } \sigma}(P \cap B(c_{\sigma}, \rho)) \setminus \sigma$.

► **Theorem 10** (Faithful reconstruction by a geometric approach). *Let $\varepsilon, \delta, \rho, \theta \geq 0$ and set $A = 4\delta\theta + 4\rho\theta^2$. Assume that $\theta \leq \frac{\pi}{6}$, $\delta \leq \varepsilon$ and $16\varepsilon \leq \rho < \frac{R}{4}$. Suppose that P is a δ -accurate ε -dense sample of \mathcal{M} that satisfies the following safety conditions:*

1. $\Theta(P, \rho) \leq \theta - 2 \arcsin\left(\frac{\rho + \delta}{R}\right)$;
2. $\text{separation}(P) > 2A + 6\delta + \frac{2\rho^2}{R}$;
3. $\text{height}(P, \rho) > 0$ and $\text{protection}(P, 3\rho) > 2A \left(1 + \frac{4d\varepsilon}{\text{height}(P, \rho)}\right)$.

Then $D = \text{Delloc}_d(P, \rho)$ enjoys the following properties:

Faithful reconstruction: D is a faithful reconstruction of \mathcal{M} ;

Circumradii: For all d -simplices $\sigma \in D$, we have that $R(\sigma) \leq \varepsilon$;

Local behaviour: For all $x \in |D|$, $\pi_{\mathbf{T}_x \mathcal{M}}(\text{St}(x, D))$ is geometrically realized.

Incidentally, under the assumption of Theorem 10, $\text{Delloc}_d(P, \rho)$ coincides $\text{FlatDel}_{\mathcal{M}}(P, \rho)$, the complex introduced and studied in the companion paper [3]. Since all the results in this paper are based on the delloc property, we find it more enlightening to formulate the results of this paper using the Delloc complex. We recall that the safety conditions can be met in practice by assuming P to be a sample of \mathcal{M} sufficiently dense and sufficiently accurate, and then perturbing the point set P as explained in the companion paper [3].

► **Remark 11.** It is easy to see that if $2R(\sigma) \leq \rho$, then a delloc simplex σ in P at scale ρ is also a *Gabriel simplex* of P , by which we mean that its smallest circumsphere $S(\sigma)$ does not enclose any point of P in its interior. In particular, if $2R(\sigma) \leq \rho$, the delloc simplex σ is a Delaunay simplex of P . Hence, under the assumptions of Theorem 10, we have the inclusion $\text{Delloc}_d(P, \rho) \subseteq \text{Del}(P)$.

4 Statement of main result

In this section, we state our main result. Hereafter, we suppose that K is a simplicial complex whose vertices are the points of P .

Orienting and signing. We also assume that \mathcal{M} together with all d -simplices of K have received an (arbitrary) orientation. For each d -simplex $\alpha \in K$ such that $\Theta(\alpha) < \frac{\pi}{2}$, we define the sign of α with respect to \mathcal{M} as follows:

$$\text{sign}_{\mathcal{M}}(\alpha) = \begin{cases} 1 & \text{if the orientation of } \alpha \text{ is consistent with that of } \mathcal{M}, \\ -1 & \text{otherwise.} \end{cases}$$

We refer the reader to the full version [2] for a formal definition of consistency and more details. We associate to any subcomplex $D \subseteq K$ the d -chain γ_D of K whose coordinates are:

$$\gamma_D(\alpha) = \begin{cases} \text{sign}_{\mathcal{M}}(\alpha) & \text{if } \alpha \in D^{[d]}, \\ 0 & \text{otherwise.} \end{cases}$$

► **Lemma 12.** *If D is a faithful reconstruction of \mathcal{M} and, for all $x \in |D|$, $\pi_{\mathbf{T}_x \mathcal{M}}(\text{St}(x, D))$ is geometrically realized, then γ_D is a cycle. In particular, this is true when $D = \text{Delloc}_d(P, \rho)$ under the assumptions of Theorem 10.*

8:8 Delaunay-Like Triangulation of Submanifolds by Minimization

Least ℓ_1 -norm problem. We define the *Delaunay energy* of the chain $\gamma \in C_d(K, \mathbb{R})$ to be its ω -weighted ℓ_1 -norm:

$$E_{\text{del}}(\gamma) = \|\gamma\|_{1,\omega} = \sum_{\alpha} \omega(\alpha) \cdot |\gamma(\alpha)| = \sum_{\alpha} \left(\int_{x \in \text{conv } \alpha} -\text{Power}_{\alpha}(x) dx \right) \cdot |\gamma(\alpha)|, \quad (1)$$

where ω is the Delaunay weight function defined in Section 2 and α ranges over all d -simplices of K . Given a d -manifold \mathcal{A} , a point $a \in \mathcal{A}$, a set of simplices $\Sigma \subseteq K$ and a d -chain γ of K , we also introduce the real number:

$$\text{load}_{a,\mathcal{A},\Sigma}(\gamma) = \sum_{\sigma \in \Sigma^{[d]}} \gamma(\sigma) \text{sign}_{\mathcal{A}}(\sigma) \mathbf{1}_{\pi_{\mathcal{A}}(\text{conv } \sigma)}(a)$$

and call it the *load* of γ on \mathcal{A} at a restricted to Σ . Letting m_0 be a generic² point on \mathcal{M} , we are interested in the following optimization problem over the set of chains in $C_d(K, \mathbb{R})$:

$$\begin{aligned} & \underset{\gamma}{\text{minimize}} && E_{\text{del}}(\gamma) \\ & \text{subject to} && \partial\gamma = 0, \\ & && \text{load}_{m_0,\mathcal{M},K}(\gamma) = 1 \end{aligned} \quad (\star)$$

Problem (\star) is a convex optimization problem and as such is solvable by linear programming. More precisely, it is a least-norm problem whose constraint functions ∂ and $\text{load}_{m_0,\mathcal{M},K}$ are clearly linear. The first constraint $\partial\gamma = 0$ expresses the fact that we are searching for d -cycles. The second constraint $\text{load}_{m_0,\mathcal{M},K}(\gamma) = 1$ can be thought of as a kind of normalization of γ . It forbids the zero chain to belong to the feasible set and we shall see that, under the assumptions of our main theorem, it forces the solution to take its coordinate values in $\{0, +1, -1\}$.

In Problem (\star) , besides the simplicial complex K that one can build from P , the knowledge of the manifold \mathcal{M} seems to be required as well for expressing the normalization constraint. What we call a *realistic* algorithm is an algorithm that takes only the point set P as input. In the full version [2], we explain how to transform Problem (\star) into an equivalent problem that does not refer to \mathcal{M} anymore, thus providing a realistic algorithm. Roughly, we replace the constraint $\text{load}_{m_0,\mathcal{M},K}(\gamma) = 1$ by a constraint of the form $\text{load}_{p_0,\Pi,\Sigma}(\gamma) = 1$, where $p_0 \in P$, Π is a d -flat that “approximates” \mathcal{M} near p_0 and Σ are simplices of K “close” to p_0 .

Main theorem. In our main theorem (see below), there is a constant $\Omega(\Delta_d)$ that depends only upon the dimension d and whose definition is given in the proof of Lemma 20.

► **Theorem 13 (Faithful reconstruction by a variational approach).** *Let ε , δ , ρ and θ be non-negative real-numbers such that $\theta \leq \frac{\pi}{6}$, $\delta \leq \varepsilon$ and $16\varepsilon \leq \rho < \frac{\mathcal{R}}{4}$. Set*

$$J = \frac{(\mathcal{R} + \rho)^d}{(\mathcal{R} - \rho)^d (\cos \theta)^{\min\{d, N-d\}}} - 1 \quad \text{and} \quad A = 4\delta\theta + 4\rho\theta^2.$$

Let $\zeta = \text{protection}(P, 3\rho)$ and suppose that P is a δ -accurate ε -dense sample of \mathcal{M} that satisfies the following safety conditions:

² Generic in the sense that it is not in the projection on \mathcal{M} of the convex hull of any $(d-1)$ -simplex of K .

1. $\Theta(P, \rho) \leq \theta - 2 \arcsin \left(\frac{\rho + \delta}{\mathcal{R}} \right)$.
2. $\text{separation}(P) > 2A + 6\delta + \frac{3\rho^2}{\mathcal{R}}$;
3. $\text{height}(P, \rho) > 0$ and $\zeta > 2A \left(1 + \frac{4d\varepsilon}{\text{height}(P, \rho)} \right)$;
4. $\zeta^2 + \zeta \text{separation}(P) > 10\rho \sin \theta (\varepsilon + \rho \sin \theta)$;
5. $J\rho^2 < (1 + J)^{-1} \frac{(d+2)(d-1)!}{4} (\zeta^2 + \zeta \text{separation}(P)) \Omega(\Delta_d)$.

Suppose that $\text{Delloc}_d(P, \rho) \subseteq K$ and that the d -simplices of K are ρ -small. Then Problem (\star) has a unique solution which is $\gamma_{\text{Delloc}_d(P, \rho)}$. The support of that solution together with all its faces coincides with $\text{Delloc}_d(P, \rho)$ and is a faithful reconstruction of \mathcal{M} .

One may ask about the feasibility of realizing the assumptions of Theorem 13. While assuming the sample to be ε -dense and δ -accurate seems realistic enough (perhaps after filtering outliers), the safety conditions seem less likely to be satisfied by natural data. In the full version [2], we show how to apply Moser Tardos Algorithm ([26] and [6, Section 5.3.4]) as a perturbation scheme to enforce the safety conditions of Theorem 13.

Choosing the simplicial complex K . Recall that the Čech complex of P at scale ρ , denoted as $\mathcal{C}(P, \rho)$, is the set of simplices of P that are ρ -small. The Rips complex of P at scale ρ , denoted as $\mathcal{R}(P, \rho)$, is a more easily-computed version which consists of all simplices of P with diameter at most 2ρ . We stress that our main theorem applies to any simplicial complex K such that $\text{Delloc}_d(P, \rho) \subseteq K \subseteq \mathcal{C}(P, \rho)$. Since $\mathcal{C}(P, r) \subseteq \mathcal{R}(P, r) \subseteq \mathcal{C}(P, \sqrt{2}r)$ and $\text{Delloc}_d(P, \rho) \subseteq \mathcal{C}(P, \varepsilon)$, it applies to any $K = \mathcal{R}(P, r)$ with $\varepsilon \leq r \leq \frac{\rho}{\sqrt{2}}$. This choice of K is well-suited for applications in high dimensional spaces. Observe that under the assumptions of Theorem 13, $\text{Delloc}_d(P, \rho) \subseteq \text{Del}(P) \cap \mathcal{C}(P, \varepsilon)$ (see Remark 11) and choosing $K = \text{Del}(P) \cap \mathcal{C}(P, r)$ for any $\varepsilon \leq r \leq \rho$ may then be more suited for applications in low dimensional spaces.

5 Technical lemma

The proof of our main theorem relies on a technical lemma which we now state and prove.

► **Lemma 14.** *Let $\mathcal{D} \subseteq \mathbb{R}^N$ be a d -manifold (with or without boundary) and K a simplicial complex with vertices in \mathbb{R}^N . Assume that there is a map $\varphi : |K| \rightarrow \mathcal{D}$. Suppose that for each d -simplex $\alpha \in K$, we have two positive weights $W(\alpha) \geq W_{\min}(\alpha)$ and that there exists a map $f : \mathcal{D} \rightarrow \mathbb{R}$ such that $W_{\min}(\alpha) = \int_{\varphi(\text{conv } \alpha)} f$. Consider the d -chain γ_{\min} on K defined by*

$$\gamma_{\min}(\alpha) = \begin{cases} 1 & \text{if } W_{\min}(\alpha) = W(\alpha), \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that $\sum_{\alpha \in K^{[d]}} \gamma_{\min}(\alpha) \mathbf{1}_{\varphi(\text{conv } \alpha)}(x) = 1$, for almost all $x \in \mathcal{D}$. Then the ℓ_1 -like norm $\|\gamma\|_{1, W}$ attains its minimum over all d -chains γ such that

$$\sum_{\alpha \in K^{[d]}} \gamma(\alpha) \mathbf{1}_{\varphi(\text{conv } \alpha)}(x) = 1, \quad \text{for almost all } x \in \mathcal{D} \tag{2}$$

if and only if $\gamma = \gamma_{\min}$.

8:10 Delaunay-Like Triangulation of Submanifolds by Minimization

Proof. We write $\tilde{\alpha} = \varphi(\text{conv } \alpha)$ throughout the proof for a shorter notation. We prove the lemma by showing that for all d -chains γ on K that satisfy constraint (2), we have:

$$\|\gamma\|_{1,W} \geq \|\gamma\|_{1,W_{\min}} \geq \int_{\mathcal{D}} f = \|\gamma_{\min}\|_{1,W_{\min}} = \|\gamma_{\min}\|_{1,W}, \quad (3)$$

with the first inequality being an equality if and only if $\gamma = \gamma_{\min}$. Clearly, $\|\gamma\|_{1,W} \geq \|\gamma\|_{1,W_{\min}}$ because $W(\alpha) \geq W_{\min}(\alpha)$. To obtain the second inequality, recall that we have assumed $\sum_{\alpha} \gamma(\alpha) \mathbf{1}_{\tilde{\alpha}}(x) = 1$ almost everywhere in \mathcal{D} . We use this to write that:

$$\|\gamma\|_{1,W_{\min}} \geq \sum_{\alpha} \gamma(\alpha) \int_{\tilde{\alpha}} f = \sum_{\alpha} \gamma(\alpha) \int_{\mathcal{D}} f \mathbf{1}_{\tilde{\alpha}} = \int_{\mathcal{D}} f \sum_{\alpha} \gamma(\alpha) \mathbf{1}_{\tilde{\alpha}} = \int_{\mathcal{D}} f, \quad (4)$$

where sums are over all d -simplices α in K . Setting $\gamma = \gamma_{\min}$ in (4), we observe that the inequality in (4) becomes an equality because none of the coefficients of γ_{\min} are negative by construction. It follows that $\int_{\mathcal{D}} f = \|\gamma_{\min}\|_{1,W_{\min}}$. Finally, $\|\gamma_{\min}\|_{1,W_{\min}} = \|\gamma_{\min}\|_{1,W}$ because γ_{\min} has been defined so that for all simplices α in its support, $W_{\min}(\alpha) = W(\alpha)$. We have thus established (3). Suppose now that $\gamma \neq \gamma_{\min}$ and let us prove that $\|\gamma\|_{1,W} > \|\gamma\|_{1,W_{\min}}$, or equivalently that

$$\sum_{\alpha \in \text{Supp } \gamma} |\gamma(\alpha)| (W(\alpha) - W_{\min}(\alpha)) > 0.$$

Since none of the terms in the above sum are negative, it suffices to show that there exists at least one simplex $\alpha \in \text{Supp } \gamma$ for which $W(\alpha) > W_{\min}(\alpha)$. By contradiction, assume that for all $\alpha \in \text{Supp } \gamma$, $W(\alpha) = W_{\min}(\alpha)$. By construction, we thus have the implication: $\gamma(\alpha) \neq 0 \implies \gamma_{\min}(\alpha) = 1$, and therefore $\text{Supp } \gamma \subseteq \text{Supp } \gamma_{\min}$. But, since $\sum_{\alpha} \gamma_{\min}(\alpha) \mathbf{1}_{\tilde{\alpha}}(x) = 1$ for almost all $x \in \mathcal{D}$ and coefficients of γ_{\min} are either 0 or 1, it follows that for almost all $x \in \mathcal{D}$, point x is covered by a unique d -simplex in the support of γ_{\min} . Hence, the simplices in $\text{Supp } \gamma_{\min}$ have pairwise disjoint interiors while their union covers \mathcal{D} . Since $\sum_{\alpha} \gamma(\alpha) \mathbf{1}_{\tilde{\alpha}}(x) = 1$ for almost all $x \in \mathcal{D}$, the simplices in $\text{Supp } \gamma$ must also cover \mathcal{D} while using only a subset of simplices in $\text{Supp } \gamma_{\min}$. The only possibility is that $\gamma = \gamma_{\min}$, yielding a contradiction. \blacktriangleleft

6 Comparing power distances

The goal of this section is to relate the two maps $\text{Power}_{\alpha}(x)$ and $\text{Power}_{\beta}(y)$ for two d -simplices $\alpha \in \text{Delloc}_d(P, \rho)$ and $\beta \subseteq P$, and for two points $x \in \text{conv } \alpha$ and $y \in \text{conv } \beta$, such that $\pi_{\mathcal{M}}(x) = \pi_{\mathcal{M}}(y)$. The main result of the section is stated in the following lemma:

► **Lemma 15.** *Let $\varepsilon, \delta, \rho \geq 0$ such that $0 \leq 2\varepsilon \leq \rho$, and $16\delta \leq \rho \leq \frac{\mathcal{R}}{3}$. Suppose that $P \subseteq \mathcal{M}^{\oplus \delta}$. Let $\zeta = \text{protection}(P, 3\rho)$ and assume that $\Theta(P, \rho) \leq \frac{\pi}{6}$, $\text{separation}(P) > \frac{3\rho^2}{\mathcal{R}} + 3\delta$ and*

$$10\rho\Theta(P, \rho) \cdot (\varepsilon + \rho\Theta(P, \rho)) < \zeta^2 + \zeta \text{separation}(P).$$

Then, for every ε -small d -simplex $\alpha \in \text{Delloc}_d(P, \rho)$, every ρ -small d -simplex $\beta \subseteq P$, every $x \in \text{conv } \alpha$, and every $y \in \text{conv } \beta$ such that $\pi_{\mathcal{M}}(x) = \pi_{\mathcal{M}}(y)$:

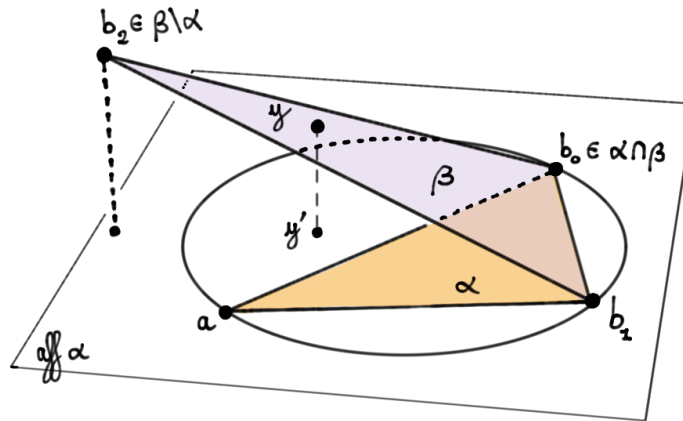
$$\text{Power}_{\beta}(y) \leq \text{Power}_{\alpha}(x) - \frac{1}{2} (\zeta^2 + \zeta \text{separation}(P)) \sum_{b \in \beta \setminus \alpha} \mu_b,$$

where $\mu_b \geq 0$ are real numbers such that $y = \sum_{b \in \beta} \mu_b b$ and $\sum_{b \in \beta} \mu_b = 1$.

To prove the lemma, we need a few auxiliary results. We start by recalling a useful expression of the power distance of a point x from the circumsphere $S(\alpha)$ of α when x is an affine combination of the vertices of α .

► **Lemma 16.** *Let $\alpha \subseteq \mathbb{R}^N$. If $x = \sum_{a \in \alpha} \lambda_a a$ with $\sum_{a \in \alpha} \lambda_a = 1$, then for every $z \in \mathbb{R}^N$*

$$\text{Power}_\alpha(x) = \|x - z\|^2 - \sum_{a \in \alpha} \lambda_a \|a - z\|^2.$$



■ **Figure 1** Notation for the proof of Lemma 17.

► **Lemma 17.** *Let α and β be two non-degenerate abstract d -simplices in \mathbb{R}^N . Suppose that $\alpha \in \text{Del}(\pi_{\text{aff } \alpha}(\alpha \cup \beta))$ and it is ζ -protected with respect to $\pi_{\text{aff } \alpha}(\alpha \cup \beta)$. Suppose furthermore that the map $\pi_{\text{aff } \alpha}|_{\alpha \cup \beta}$ is injective. Then for every convex combination $y = \sum_{b \in \beta} \mu_b b$ with $\mu_b \geq 0$ and $\sum_{b \in \beta} \mu_b = 1$, we have*

$$\text{Power}_\beta(y) \leq \text{Power}_\alpha(\pi_{\text{aff } \alpha}(y)) - (\zeta^2 + 2\zeta R(\alpha)) \sum_{b \in \beta \setminus \alpha} \mu_b.$$

Proof. See Figure 1. Let $Z(\alpha)$ be the radius of the $(d - 1)$ -dimensional circumsphere of α . Clearly, $\|a - Z(\alpha)\| = R(\alpha)$ for all $a \in \alpha$. Let $Q = \pi_{\text{aff } \alpha}(\alpha \cup \beta)$. Since $\alpha \in \text{Del}(Q)$ and is ζ -protected with respect to Q , we get:

$$\begin{aligned} (R(\alpha) + \zeta)^2 &< \|\pi_{\text{aff } \alpha}(b) - Z(\alpha)\|^2, & \text{for all } b \in \beta \setminus \alpha, \\ R(\alpha)^2 &= \|\pi_{\text{aff } \alpha}(b) - Z(\alpha)\|^2, & \text{for all } b \in \beta \cap \alpha. \end{aligned}$$

Multiplying both sides of each equation above by μ_b and summing over all $b \in \beta$, we obtain:

$$R(\alpha)^2 + (\zeta^2 + 2\zeta R(\alpha)) \sum_{b \in \beta \setminus \alpha} \mu_b \leq \sum_{b \in \beta} \mu_b \|\pi_{\text{aff } \alpha}(b) - Z(\alpha)\|^2. \tag{5}$$

For short, write $y' = \pi_{\text{aff } \alpha}(y)$ and $\beta' = \pi_{\text{aff } \alpha}(\beta)$. Noting that $y' = \sum_{b \in \beta} \mu_b b'$ and applying Lemma 16 with $z = Z(\alpha)$, we get that

$$\text{Power}_{\beta'}(y') = \|y' - Z(\alpha)\|^2 - \sum_{b \in \beta} \mu_b \|\pi_{\text{aff } \alpha}(b) - Z(\alpha)\|^2.$$

8:12 Delaunay-Like Triangulation of Submanifolds by Minimization

Subtracting $\|y' - Z(\alpha)\|^2$ from both sides of (5) and using the above expression, we obtain

$$-\text{Power}_\alpha(y') + (\zeta^2 + 2\zeta R(\alpha)) \sum_{b \in \beta \setminus \alpha} \mu_b \leq -\text{Power}_{\beta'}(y').$$

Applying Lemma 16 again, with $Z = y'$ and $Z = y$ respectively, we get that:

$$-\text{Power}_{\beta'}(y') = \sum_{b \in \beta} \mu_b \|\pi_{\text{aff } \alpha}(b) - \pi_{\text{aff } \alpha}(y)\|^2 \leq \sum_{b \in \beta} \mu_b \|b - y\|^2 = -\text{Power}_\beta(y),$$

which concludes the proof. \blacktriangleleft

► **Lemma 18.** *Let α and β be two non-degenerate abstract d -simplices in \mathbb{R}^N . Suppose that $\alpha \in \text{Del}(\pi_{\text{aff } \alpha}(\alpha \cup \beta))$ and α is ζ -protected with respect to $\pi_{\text{aff } \alpha}(\alpha \cup \beta)$. Suppose that the map $\pi_{\text{aff } \alpha}|_{\alpha \cup \beta}$ is injective and that both $\text{conv } \alpha$ and $\text{conv } \beta$ are contained in the ρ -tubular neighborhood of \mathcal{M} . Suppose furthermore that β is ρ -small. If $\Theta(\alpha) < \frac{\pi}{6}$ and*

$$5\rho \sin \Theta(\alpha) \cdot (2R(\alpha) + 2\rho \sin \Theta(\alpha)) < \zeta^2 + 2\zeta R(\alpha),$$

then for every $x \in \text{conv } \alpha$ and every $y \in \text{conv } \beta$ with $\pi_{\mathcal{M}}(x) = \pi_{\mathcal{M}}(y)$, we have

$$\text{Power}_\beta(y) \leq \text{Power}_\alpha(x) - \frac{1}{2}(\zeta^2 + 2\zeta R(\alpha)) \sum_{b \in \beta \setminus \alpha} \mu_b,$$

where $\mu_b \geq 0$ are real numbers such that $y = \sum_{b \in \beta} \mu_b b$ and $\sum_{b \in \beta} \mu_b = 1$.

7 Proving the main result

Suppose that K is a simplicial complex with vertex set P . Write $D = \text{Delloc}_d(P, \rho)$, $\mathcal{D} = |D|$ and $\mathcal{K} = |K|$ for short. In this section, we prove our main theorem by applying Lemma 14. This requires us to define two maps $\varphi : \mathcal{K} \rightarrow \mathcal{D}$ and $f : \mathcal{D} \rightarrow \mathbb{R}$, two weights $W(\alpha)$ and $W_{\min}(\alpha)$ for each d -simplex $\alpha \in K$, and to check that these maps and weights satisfy the requirements of Lemma 14. For each $\alpha \in K$, let $W(\alpha) = \omega(\alpha)$ be the Delaunay weight of α . To be able to define φ , f , and W_{\min} , we assume that the following conditions are met:

- (1) D is a faithful reconstruction of \mathcal{M} ;
- (2) For every d -simplex $\sigma \subseteq K$, the map $\pi_{\mathcal{M}}|_{\text{conv } \sigma}$ is well-defined and injective.

These conditions are easily derived from the assumptions of the main theorem. We are now ready to introduce additional notation. Consider a subset $X \subseteq \mathbb{R}^N$ and suppose that the map $\pi_{\mathcal{M}}|_X$ is well-defined and injective. Then it is possible to define a bijective map $\pi_{X \rightarrow \mathcal{M}} : X \rightarrow \pi_{\mathcal{M}}(X)$. Because D is a faithful reconstruction of \mathcal{M} , the map $\pi_{\mathcal{D} \rightarrow \mathcal{M}}$ is well-defined and bijective. Similarly, for every d -simplex $\sigma \in K$, the map $\pi_{\text{conv } \sigma \rightarrow \mathcal{M}}$ is well-defined and bijective. We now introduce the map $\varphi : \mathcal{K} \rightarrow \mathcal{D}$ defined by $\varphi = [\pi_{\mathcal{D} \rightarrow \mathcal{M}}]^{-1} \circ \pi_{\mathcal{M}}$ and let $f : \mathcal{D} \rightarrow \mathbb{R}$ be the map defined by:

$$f(x) = \min_{\sigma} \left(-\text{Power}_\sigma([\pi_{\text{conv } \sigma \rightarrow \mathcal{M}}]^{-1} \circ \pi_{\mathcal{M}}(x)) \right), \quad (6)$$

where the minimum is taken over all d -simplices $\sigma \in K$ such that $x \in \varphi(\text{conv } \sigma)$. Note that $f(x)$ can be defined equivalently as the minimum of $-\text{Power}_\beta(y)$ over all d -simplices $\beta \in K$ and all points $y \in \text{conv } \beta$ such that $\pi_{\mathcal{M}}(x) = \pi_{\mathcal{M}}(y)$. Given a d -simplex $\sigma \in K$, we associate to σ the weight:

$$W_{\min}(\sigma) = \int_{x \in \varphi(\text{conv } \sigma)} f(x) dx. \quad (7)$$

► **Lemma 19.** *Under the assumptions of Theorem 13:*

- For every d -simplex $\alpha \in D$ and every point $x \in \text{conv } \alpha$, we have $f(x) = -\text{Power}_\alpha(x)$.
- For every d -simplex $\alpha \in D$, we have $W_{\min}(\alpha) = W(\alpha)$.

Proof. Consider a d -simplex $\alpha \in D$, a d -simplex $\beta \in K$, $x \in \text{conv } \alpha$ and $y \in \text{conv } \beta$ such that $\pi_{\mathcal{M}}(x) = \pi_{\mathcal{M}}(y)$. Applying Lemma 15, we obtain that $\text{Power}_\beta(y) \leq \text{Power}_\alpha(x)$ or equivalently $\text{Power}_\beta([\pi_{\text{conv } \beta \rightarrow \mathcal{M}}]^{-1} \circ \pi_{\mathcal{M}}(x)) \leq \text{Power}_\alpha(x)$ and therefore $f(x) = -\text{Power}_\alpha(x)$. To establish the second item of the lemma, notice that for all $\alpha \in D$, the restriction of φ to $\text{conv } \alpha$ is the identity function, $\varphi|_{\text{conv } \alpha} = \text{Id}$ and therefore $\varphi(\text{conv } \alpha) = \text{conv } \alpha$. Since we have just established that $f(x) = -\text{Power}_\alpha(x)$, we get that

$$W_{\min}(\alpha) = \int_{x \in \varphi(\text{conv } \alpha)} f(x) dx = \int_{x \in \text{conv } \alpha} -\text{Power}_\alpha(x) dx = \omega(\alpha) = W(\alpha),$$

which concludes the proof. ◀

► **Lemma 20.** *Under the assumptions of Theorem 13, for every d -simplex $\beta \in K \setminus D$, we have $W_{\min}(\beta) < W(\beta)$.*

Proof. We need some notation. Given α and β in K , we write $\text{conv}_{|\alpha} \beta$ for the set of points $y \in \text{conv } \beta$ for which there exists a point $x \in \text{conv } \alpha$ such that $\pi_{\mathcal{M}}(x) = \pi_{\mathcal{M}}(y)$. We define the map $\varphi_{\beta \rightarrow \alpha} : \text{conv}_{|\alpha} \beta \rightarrow \text{conv}_{|\beta} \alpha$ as $\varphi_{\beta \rightarrow \alpha}(y) = [\pi_{\text{conv } \alpha \rightarrow \mathcal{M}}]^{-1} \circ \pi_{\text{conv } \beta \rightarrow \mathcal{M}}(y)$. Note that $\varphi_{\beta \rightarrow \alpha}$ is invertible and its inverse is $\varphi_{\alpha \rightarrow \beta}$. Also, note that J in Theorem 13 has been chosen precisely so that one can apply Lemma 38 in [2] and guarantee that $|\det(J\varphi_{\beta \rightarrow \alpha})(y)| \in [\frac{1}{1+J}, 1+J]$ for all $\alpha, \beta \in K$ and all $y \in \text{conv}_{|\alpha} \beta$. Consider a d -simplex $\beta \in K \setminus D$. By Lemma 19, $f(x) = -\text{Power}_\alpha(x)$ and therefore:

$$W_{\min}(\beta) = \sum_{\alpha \in D^{[d]}} \int_{x \in \text{conv}_{|\beta} \alpha} -\text{Power}_\alpha(x) dx.$$

For any convex combination y of points in β , let $\{\mu_b^\beta(y)\}_{b \in \beta}$ designate the family of non-negative real numbers summing up to 1 such that $y = \sum_{b \in \beta} \mu_b^\beta(y)b$. Plugging in the upper bound on $-\text{Power}_\alpha(x)$ provided by Lemma 15, letting

$$c = \frac{1}{2} (\zeta^2 + \zeta \text{ separation}(P)),$$

and making the change of variable $x = \varphi_{\beta \rightarrow \alpha}(y)$, we upper bound $W_{\min}(\beta)$ as follows:

$$\begin{aligned} W_{\min}(\beta) &\leq \sum_{\alpha \in D^{[d]}} \int_{x \in \text{conv}_{|\beta} \alpha} \left[-\text{Power}_\beta(\varphi_{\alpha \rightarrow \beta}(x)) - c \sum_{b \in \beta \setminus \alpha} \mu_b^\beta(\varphi_{\alpha \rightarrow \beta}(x)) \right] dx \\ &= \sum_{\alpha \in D^{[d]}} \int_{y \in \text{conv}_{|\alpha} \beta} \left[-\text{Power}_\beta(y) - c \sum_{b \in \beta \setminus \alpha} \mu_b^\beta(y) \right] |\det(J\varphi_{\beta \rightarrow \alpha})(y)| dy \\ &\leq (1+J)W(\beta) - (1+J)^{-1}c \sum_{\alpha \in D^{[d]}} \int_{y \in \text{conv}_{|\alpha} \beta} \sum_{b \in \beta \setminus \alpha} \mu_b^\beta(y) dy. \end{aligned}$$

A key observation is that, because $\beta \neq \alpha$, then $\beta \setminus \alpha \neq \emptyset$. Therefore the sum $\sum_{b \in \beta \setminus \alpha} \mu_b^\beta(y)$ is always lower bounded by $\inf_{b \in \beta} \mu_b^\beta(y)$. Associating the quantity

$$\Omega(\beta) = \int_{y \in \text{conv } \beta} \inf_{b \in \beta} \mu_b^\beta(y) dy,$$

8:14 Delaunay-Like Triangulation of Submanifolds by Minimization

to β we thus obtain that $W_{\min}(\beta) \leq (1+J)W(\beta) - (1+J)^{-1}c\Omega(\beta)$. Hence, $W_{\min}(\beta) < W(\beta)$ as long as

$$JW(\beta) < (1+J)^{-1}c\Omega(\beta). \quad (8)$$

Using a change of variable, it is not too difficult to show that $\Omega(\beta) = d! \text{vol}(\beta)\Omega(\Delta_d)$, where $\Delta_d = \{\lambda \in \mathbb{R}^d \mid \sum_{i=1}^d \lambda_i \leq 1; \lambda_i \geq 0, i = 1, 2, \dots, d\}$ represents the standard d -simplex. Remark that $\Omega(\Delta_d)$ is a constant that depends only upon the dimension d and is thus universal. Plugging in $\Omega(\beta) = d! \text{vol}(\beta)\Omega(\Delta_d)$ on the right side of (8), and the expression of $W(\beta) = \omega(\beta)$ given by Lemma 6 on the left side of (8), and recalling that β is ρ -small, we find that condition (8) is implied by the following condition:

$$J\rho^2 < (1+J)^{-1} \frac{(d+2)(d-1)!}{4} (\zeta^2 + \zeta \text{separation}(P)) \Omega(\Delta_d),$$

which we have assumed to hold. \blacktriangleleft

Proof of Theorem 13. We start with pointing out that Problem (\star) is invariant under change of orientation of d -simplices in K and thus we may assume that every d -simplex α in K has an orientation that is consistent with that of \mathcal{M} , that is, $\text{sign}_{\mathcal{M}}(\alpha) = 1$ for all $\alpha \in K^{[d]}$. Let $D = \text{Deloc}_d(P, \rho)$, $\mathcal{D} = |D|$ and $\mathcal{K} = |K|$. Theorem 10 ensures that \mathcal{D} is a d -manifold and $\pi_{\mathcal{M}} : \mathcal{D} \rightarrow \mathcal{M}$ is a homeomorphism. Define $\varphi : \mathcal{K} \rightarrow \mathcal{D}$, $f : \mathcal{D} \rightarrow \mathbb{R}$, W , and W_{\min} as explained at the beginning of the section. Consider the d -chain γ_{\min} on K :

$$\gamma_{\min}(\alpha) = \begin{cases} 1 & \text{if } W_{\min}(\alpha) = W(\alpha), \\ 0 & \text{otherwise.} \end{cases}$$

By Lemma 19 and Lemma 20, the following property holds: for all $\alpha \in K$, $W_{\min}(\alpha) = W(\alpha)$ if and only if α is a d -simplex of D . It follows that $\gamma_{\min} = \gamma_D$. Furthermore, we have $\sum_{\alpha \in K^{[d]}} \gamma_{\min}(\alpha) \mathbf{1}_{\varphi(\text{conv } \alpha)}(x) = \sum_{\alpha \in D^{[d]}} \mathbf{1}_{\text{conv } \alpha}(x) = 1$ for almost all $x \in \mathcal{D}$. Recalling that $W = \omega$ and therefore $\|\gamma\|_{1,W} = E_{\text{del}}(\gamma)$, and applying Lemma 14, we deduce that $\gamma_{\min} = \gamma_D$ is the unique solution to the following optimization problem over the set of chains in $C_d(K, \mathbb{R})$:

$$\begin{aligned} & \underset{\gamma}{\text{minimize}} && E_{\text{del}}(\gamma) \\ & \text{subject to} && \sum_{\alpha \in K^{[d]}} \gamma(\alpha) \text{sign}_{\mathcal{M}}(\alpha) \mathbf{1}_{\varphi(\text{conv } \alpha)}(x) = 1, \text{ for almost all } x \in \mathcal{D} \quad (\star\star) \end{aligned}$$

One can see that Problem $(\star\star)$ remains unchanged if one replaces the constraint with

$$\sum_{\alpha \in K^{[d]}} \gamma(\alpha) \text{sign}_{\mathcal{M}}(\alpha) \mathbf{1}_{\pi_{\mathcal{M}}(\text{conv } \alpha)}(m) = 1, \quad \text{for almost all } m \in \mathcal{M}. \quad (9)$$

Let m_0 be the arbitrary generic point of \mathcal{M} , as in Problem (\star) . By Lemma 48 in [2], the above constraint is equivalent to the following set of constraints:

$$\begin{cases} \partial\gamma = 0, \\ \sum_{\alpha \in K^{[d]}} \gamma(\alpha) \text{sign}_{\mathcal{M}}(\alpha) \mathbf{1}_{\pi_{\mathcal{M}}(\text{conv } \alpha)}(m_0) = 1. \end{cases}$$

We deduce that Problem (\star) and Problem $(\star\star)$ are equivalent, and we get the result. \blacktriangleleft

References

- 1 Pierre Alliez, David Cohen-Steiner, Mariette Yvinec, and Mathieu Desbrun. Variational tetrahedral meshing. *ACM Transactions on Graphics (TOG)*, 24(3):617–625, 2005.
- 2 D. Attali and A. Lieutier. Delaunay-like triangulation of smooth orientable submanifolds by ℓ_1 -norm minimization, 2022. [arXiv:2203.06008](https://arxiv.org/abs/2203.06008).
- 3 D. Attali and A. Lieutier. Flat delaunay complexes for homeomorphic manifold reconstruction, 2022. [arXiv:2203.05943](https://arxiv.org/abs/2203.05943).
- 4 D. Attali, A. Lieutier, and D. Salinas. Vietoris–rips complexes also provide topologically correct reconstructions of sampled shapes. *Computational Geometry: Theory and Applications*, 46(4):448–465, 2013.
- 5 J.-D. Boissonnat and A. Ghosh. Manifold reconstruction using tangential delaunay complexes. *Discrete & Computational Geometry*, 51(1):221–267, 2014.
- 6 Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and topological inference*, volume 57. Cambridge University Press, 2018.
- 7 Jean-Daniel Boissonnat, Ramsay Dyer, and Arijit Ghosh. The stability of delaunay triangulations. *International Journal of Computational Geometry & Applications*, 23(04n05):303–333, August 2013. doi:10.1142/s0218195913600078.
- 8 Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic geometry*. Cambridge university press, 1998.
- 9 Glencora Borradaile, William Maxwell, and Amir Nayyeri. Minimum Bounded Chains and Minimum Homologous Chains in Embedded Simplicial Complexes. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2020.21.
- 10 Erin W. Chambers, Jeff Erickson, and Amir Nayyeri. Minimum cuts and shortest homologous cycles. In *Proceedings of the 25th Annual Symposium on Computational Geometry - SCG '09*, page 377, Aarhus, Denmark, 2009. ACM Press. doi:10.1145/1542362.1542426.
- 11 F. Chazal, D. Cohen-Steiner, and A. Lieutier. A sampling theory for compact sets in Euclidean space. *Discrete and Computational Geometry*, 41(3):461–479, 2009.
- 12 F. Chazal and A. Lieutier. Smooth Manifold Reconstruction from Noisy and Non Uniform Approximation with Guarantees. *Computational Geometry: Theory and Applications*, 40:156–170, 2008.
- 13 Chao Chen and Daniel Freedman. Hardness Results for Homology Localization. *Discrete & Computational Geometry*, 45(3):425–448, April 2011. doi:10.1007/s00454-010-9322-8.
- 14 L. Chen and M. Holst. Efficient mesh optimization schemes based on optimal delaunay triangulations. *Computer Methods in Applied Mechanics and Engineering*, 200(9):967–984, 2011.
- 15 Long Chen. Mesh smoothing schemes based on optimal delaunay triangulations. In *Proceedings of the 13th International Meshing Roundtable, IMR 2004, Williamsburg, Virginia, USA, September 19-22, 2004*, pages 109–120, 2004. URL: <http://imr.sandia.gov/papers/abstracts/Ch317.html>.
- 16 Long Chen and Jin-chao Xu. Optimal delaunay triangulations. *Journal of Computational Mathematics*, pages 299–308, 2004.
- 17 Zhonggui Chen, Wenping Wang, Bruno Lévy, Ligang Liu, and Feng Sun. Revisiting optimal delaunay triangulation for 3d graded mesh generation. *SIAM Journal on Scientific Computing*, 36(3):A930–A954, 2014.
- 18 David Cohen-Steiner, André Lieutier, and Julien Vuillamy. Regular triangulations as lexicographic optimal chains. *Preprint HAL*, 2019. URL: <https://hal.archives-ouvertes.fr/hal-02391285>.

- 19 Tamal K. Dey, Anil N. Hirani, and Bala Krishnamoorthy. Optimal Homologous Cycles, Total Unimodularity, and Linear Programming. *SIAM Journal on Computing*, 40(4):1026–1044, January 2011. doi:10.1137/100800245.
- 20 Tamal K. Dey, Tao Hou, and Sayan Mandal. Computing minimal persistent cycles: Polynomial and hard cases. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '20*, pages 2587–2606, USA, January 2020. Society for Industrial and Applied Mathematics.
- 21 H. Edelsbrunner. *Geometry and topology for mesh generation*. Cambridge Univ Pr, 2001.
- 22 Herbert Edelsbrunner and Nimish R Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15(3):223–241, 1996.
- 23 H. Federer. Curvature measures. *Trans. Amer. Math. Soc*, 93:418–491, 1959.
- 24 Jisu Kim, Jaehyeok Shin, Frédéric Chazal, Alessandro Rinaldo, and Larry Wasserman. Homotopy reconstruction via the cech complex and the vietoris-rips complex. *arXiv preprint*, 2019. arXiv:1903.06955.
- 25 Frank Morgan. *Geometric measure theory: a beginner's guide*. Academic press, 2016.
- 26 Robin A Moser and Gábor Tardos. A constructive proof of the general lovász local lemma. *Journal of the ACM (JACM)*, 57(2):1–15, 2010.
- 27 J.R. Munkres. *Elements of algebraic topology*. Perseus Books, 1993.
- 28 Oleg R Musin. Properties of the delaunay triangulation. In *Proceedings of the thirteenth annual symposium on Computational geometry*, pages 424–426, 1997.
- 29 P. Niyogi, S. Smale, and S. Weinberger. Finding the Homology of Submanifolds with High Confidence from Random Samples. *Discrete Computational Geometry*, 39(1-3):419–441, 2008.
- 30 Samuel Rippa. Minimal roughness property of the delaunay triangulation. *Computer Aided Geometric Design*, 7(6):489–497, 1990.
- 31 Julien Vuillamy. *Simplification planimétrique et chaînes lexicographiques pour la reconstruction 3D de scènes urbaines*. PhD thesis, Université Côte d'Azur, 2021.

Tighter Bounds for Reconstruction from ϵ -Samples

Håvard Bakke Bjerkevik  

Institute of Geometry, Technische Universität Graz, Austria

Abstract

We show that reconstructing a curve in \mathbb{R}^d for $d \geq 2$ from a 0.66-sample is always possible using an algorithm similar to the classical NN-CRUST algorithm. Previously, this was only known to be possible for 0.47-samples in \mathbb{R}^2 and $\frac{1}{3}$ -samples in \mathbb{R}^d for $d \geq 3$. In addition, we show that there is not always a unique way to reconstruct a curve from a 0.72-sample; this was previously only known for 1-samples. We also extend this non-uniqueness result to hypersurfaces in all higher dimensions.

2012 ACM Subject Classification Mathematics of computing \rightarrow Geometric topology

Keywords and phrases Curve reconstruction, surface reconstruction, ϵ -sampling

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.9

Related Version *Full Version:* <https://arxiv.org/abs/2112.03656v2> [8]

Funding The author is supported by the Austrian Science Fund (FWF) grant number P 33765-N.

Acknowledgements The author would like to thank Michael Kerber for insights into the size of Delaunay triangulations, and Stefan Ohrhallinger and Scott A. Mitchell for answering my questions about the state of the art of curve and surface reconstruction.

1 Introduction

The main problem considered in this paper is that of *curve reconstruction*. Given a (finite) set of points \mathcal{S} in \mathbb{R}^d , we assume that this is a subset of a union \mathcal{C} of closed curves, and we want to reconstruct \mathcal{C} knowing only \mathcal{S} . Reconstructing \mathcal{C} exactly from a finite set of points is unfeasible, so we restrict the problem to finding the graph $G_{\mathcal{C}}(\mathcal{S})$ on \mathcal{S} induced by \mathcal{C} : there is an edge in $G_{\mathcal{C}}(\mathcal{S})$ between two points in \mathcal{S} if you can walk from one to the other along \mathcal{C} without meeting another point of \mathcal{S} .

To do this, one needs an assumption on \mathcal{S} and \mathcal{C} . Some work on curve reconstruction and similar problems uses global assumptions for instance related to the maximum curvature [5, 7, 12, 21, 24, 25]. A weakness of this approach is that it may force you to sample the whole curve densely even if just a small portion of it has large curvature. An influential paper by Amenta, Bern and Eppstein [3] introduced the CRUST algorithm along with a local sampling condition allowing the sampling density to vary depending on the local distance to the *medial axis* of \mathcal{C} . To be precise, they guarantee correct reconstruction for any ϵ -sampled curve in the plane whenever $\epsilon < 0.252$. The condition that a curve is ϵ -sampled is weaker the larger ϵ is, so we would like to guarantee correct reconstruction for ϵ -sampled curves for as large an ϵ as possible.

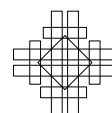
There followed a number of papers seeking to improve the sampling conditions of [3]: Dey and Kumar [15] introduced NN-CRUST (NN = nearest neighbor), which allows curves in higher-dimensional space, and prove that correct reconstruction is guaranteed for $\epsilon < \frac{1}{3}$; Lenz [20] defines a family of algorithms of which NN-CRUST is a special case and conjectures that $\epsilon \leq 0.48$ is sufficient for correctness in another special case; and Ohrhallinger et al. [22] introduce HNN-CRUST, proving correct reconstruction for $\epsilon < 0.47$, and also for $\rho < 0.9$, where ρ is a *reach*-based parameter that is related to (but different from) the parameter ϵ . It is shown in [3, Observation 6] that correct reconstruction cannot be guaranteed for $\epsilon \geq 1$.



© Håvard Bakke Bjerkevik;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 9; pp. 9:1–9:17
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



In addition, there have been several papers improving on [3] in other ways: Gold [18] simplified the CRUST algorithm; Dey et al. [16] gave an algorithm allowing open curves; and Dey and Wenger [17] considered curves with corners. Finally, we mention that [2] ties the ϵ -sampling condition to a completely different approach to curve reconstruction by showing that a solution of the traveling salesman problem on the sample points gives a correct reconstruction from an ϵ -sample for $\epsilon < 0.1$. For further references, we refer to the recent survey of Ohrhallinger et al. [23] on curve reconstruction in the plane.

Moving up to higher dimensions, one can consider the problem of *submanifold reconstruction* [1, 10, 13, 14, 21]. Instead of working with samples of a curve, one assumes that the points are sampled from a submanifold in \mathbb{R}^d for $d \geq 3$; the case of surfaces in \mathbb{R}^3 is of particular interest. While this is not the main focus of the paper, we note that this problem is important from a practical point of view; see for instance [6] for a survey covering the literature related to 3D scanings with imperfections. So far, the results using ϵ -sampling have been much weaker for surface reconstruction than for curve reconstruction. For $d = 3$, correct surface reconstruction is only known to be possible to guarantee for $\epsilon \leq 0.06$ [4].

1.1 Our contributions

The question we study is: For which ϵ is it possible to guarantee correct curve reconstruction using an ϵ -sample? Despite the popularity of ϵ -sampling as a sampling condition in the literature and the body of work aiming to weaken sampling conditions, there is still a large gap between the ϵ for which we know that reconstruction is always possible and the ϵ for which we know that it is not always possible: For any $\epsilon \in (0.47, 1)$, it is as far as the author knows an open question if it is possible to guarantee correct reconstruction of a curve (or union of curves) in \mathbb{R}^2 using an ϵ -sample. For curves in \mathbb{R}^d , $d \geq 3$, the same is true for $\epsilon \in (\frac{1}{3}, 1)$. We improve this situation drastically in both ends. First we describe algorithms that guarantee correct reconstruction for $\epsilon = 0.66$ for all $d \geq 2$. Algorithm 1 runs in $O(n^2)$ for any fixed d , and Algorithm 2 runs in $O(n \log n)$ for $d = 2$. While we have not implemented our algorithms, we believe that the speed of Algorithm 2 in practice is comparable to that of the algorithms in [15] and [22] because of their similarities.

Secondly, we give an example demonstrating that one cannot in general guarantee correct reconstruction using 0.72-samples for any $d \geq 2$. Thus, the interval of ϵ for which it is unknown if an ϵ -sample is enough for reconstruction is reduced from $(0.47, 1)$ (or $(\frac{1}{3}, 1)$ for $d \geq 3$) to $(0.66, 0.72)$.

By a straightforward generalization, we use our example to prove that a 0.72-sample is not in general enough to guarantee correct reconstruction of a manifold of any dimension. We do not show any positive results in higher dimensions, but we hope that since we do not put any restriction on the ambient dimension of the set of samples, our ideas can be useful also for reconstruction of higher-dimensional manifolds.

A serious alternative to the ϵ -sampling condition is the ρ -sampling condition of [22]. The authors of [22] argue that ϵ -sampling with $\epsilon \leq 0.47$ requires more sample points than what ρ -sampling does. With our new bounds on ϵ , the situation changes somewhat. An in-depth discussion of the relationship between ϵ -sampling and ρ -sampling is beyond the scope of this paper (as is the question of whether the two sampling conditions can be combined in a way that exploits the advantages of both of them), but we study some instructive examples in the full version of the paper [8, Appendix B]. To summarize, ρ -sampling seems to do better for curves with slowly changing curvature, while ϵ does better in some examples with rapidly changing curvature. Both our upper and lower bounds for ϵ help us understand the relative strengths of ϵ - and ρ -sampling.

We begin by introducing necessary definitions and notation in Section 2, before we prove the main theorem in Section 3. In Section 4, we show that correct reconstruction from 0.72-samples is not always possible, and we finish off by generalizing the example to higher dimensions in Section 5.

2 Definitions and notation

Throughout most of the paper, we work with a finite, disconnected union \mathcal{C} of closed curves in \mathbb{R}^d for some fixed $d \geq 2$, and a finite subset \mathcal{S} of \mathcal{C} . We will call the elements of \mathcal{S} *sample points*. By a closed curve, we mean the image of an injective map from the circle. Sometimes it will be convenient to fix an orientation of (a connected component of) \mathcal{C} . The notation $a \rightarrow b$ means that we have chosen an orientation of a connected component of \mathcal{C} containing $a, b \in \mathcal{S}$ and that by starting at a and moving along \mathcal{C} following this orientation, the next element of \mathcal{S} one encounters is b . We use the shorthand $a \rightarrow b \rightarrow c$ when we mean $a \rightarrow b$ and $b \rightarrow c$. For p, q in the same connected component of \mathcal{C} , we define $[p, q]$ as $\{p\}$ if $p = q$, and as the image of any injective path from p to q that is consistent with the orientation of \mathcal{C} if $p \neq q$. We define $[a, b)$, $(a, b]$ and (a, b) similarly depending on whether a and/or b are included or not. By a *midpoint of* $[a, b]$ we mean a point $p \in [a, b]$ with $d(p, a) = d(p, b)$, where $d(x, y)$ denotes Euclidean distance.

If $(a \rightarrow) b \rightarrow c$ or $c \rightarrow b(\rightarrow a)$, we say that $(a,)$ b and c are *consecutive*. We define $G_{\mathcal{C}}(\mathcal{S})$ as the graph on \mathcal{S} with an edge between a and b if and only if a and b are consecutive.

For $X \subset \mathbb{R}^d$, let $d(x, X) := \inf_{y \in X} d(x, y)$. The *medial axis* \mathcal{M} [9] is the set of points in \mathbb{R}^d that do not have a unique closest point in \mathcal{C} . For $p \in \mathcal{C}$, the *local feature size* $\text{lfs}(p)$ is defined as $d(p, \mathcal{M})$. For $\epsilon > 0$, we say that $\mathcal{S} \subset \mathcal{C}$ is an ϵ -*sample* (of \mathcal{C}) if for all $p \in \mathcal{C}$, $d(p, \mathcal{S}) < \epsilon \text{lfs}(p)$. Note that being an ϵ -sample is a stronger condition the smaller ϵ is. Throughout the paper we will assume that \mathcal{S} is an ϵ -sample, but our assumptions on ϵ will vary.

We define $\text{cl}: \mathbb{R}^d \setminus \mathcal{M} \rightarrow \mathcal{C}$ by letting $\text{cl}(x)$ be the point in \mathcal{C} closest to x ; i.e., $\text{cl}(x) = \arg \min_{p \in \mathcal{C}} d(x, p)$. It follows immediately from the definition of \mathcal{M} that cl is well-defined. We prove that cl is continuous in Lemma 2.

We use the notation $B_x(r)$ for the closed ball with radius r centered at $x \in \mathbb{R}^d$. For $x, y \in \mathbb{R}^d$, the closed line segment from x to y is denoted by \overline{xy} .

We often restrict our attention to a plane $\Pi \subset \mathbb{R}^d$, which we identify with \mathbb{R}^2 . This way, we can associate canonical coordinates (x, y) to each point $p \in \Pi$.

3 Proof that 0.66-samples allow reconstruction

This section is devoted to giving a proof of the main theorem:

► **Theorem 1.** *Let \mathcal{C} be a union of closed curves in \mathbb{R}^d for some $d \geq 2$, and let \mathcal{S} be a 0.66-sample of \mathcal{C} containing n points. Given \mathcal{S} as input, NN-COMPATIBLE and COMPATIBLE-CRUST both compute $G_{\mathcal{C}}(\mathcal{S})$. The former runs in $O(n^2)$, and for $d = 2$, the latter runs in $O(n \log n)$.*

The algorithms are rather simple, and are similar to the previous CRUST-type algorithms. To be specific, COMPATIBLE-CRUST borrows the idea from [3] of only selecting edges from the Delaunay triangulation¹, and both algorithms use the idea from [15] of including an edge

¹ For an introduction to Delaunay triangulations in the plane, see [11, Chapter 9].

9:4 Tighter Bounds for Reconstruction from ϵ -Samples

between each sample point and its nearest neighbor (called “closest” in the algorithms) in addition to the nearest neighbor satisfying some condition related to the angle between the resulting two edges (called “clComp” in the algorithms). The new ingredient in our algorithm is that we require triples of consecutive points to be *compatible* (see Figure 3), which is a different criterion than those used in previous algorithms. We define this compatibility property in Section 3.3. This criterion has the advantage over criteria used in previous papers in that it is the optimal local criterion for when a triple of points can be consecutive: If a triple is not compatible, it cannot be consecutive, while if it is compatible, there is a curve passing through the three points that does not violate the sampling condition locally. It will be clear from the definition that checking if a triple $(a, b, c) \in \mathcal{S}^3$ is compatible can be done in constant time. The separation into two algorithms is done to optimize the running time: For $d = 2$, computing the Delaunay triangulation saves us time, while for $d \geq 3$, a more straightforward approach is at least as efficient in the worst case.

■ Algorithm 1 NN-COMPATIBLE.

Input: 0.66-sample $\mathcal{S} \subset \mathbb{R}^d$ of \mathcal{C} for $d \geq 2$
Output: $G_{\mathcal{C}}(\mathcal{S})$
Initialize $G \leftarrow \{\}$
foreach $x \in \mathcal{S}$ **do**
 closest $\leftarrow \arg \min_{y \in \mathcal{S} \setminus \{x\}} \{d(x, y)\}$
 CompNeigh $\leftarrow \{y \in \mathcal{S} \mid (\text{closest}, x, y) \text{ is compatible}\}$
 clComp $\leftarrow \arg \min_{y \in \text{CompNeigh}} \{d(x, y)\}$
 $G \leftarrow G \cup \{x, \text{closest}\}, \{x, \text{clComp}\}$
return G

In NN-COMPATIBLE, we run through the for-loop n times. Each line in the loop can be executed in $O(n)$, which gives a total running time of $O(n^2)$.

■ Algorithm 2 COMPATIBLE-CRUST.

Input: 0.66-sample $\mathcal{S} \subset \mathbb{R}^d$ of \mathcal{C} for $d \geq 2$
Output: $G_{\mathcal{C}}(\mathcal{S})$
Compute the 1-skeleton $D_1(\mathcal{S})$ of a Delaunay triangulation of \mathcal{S} .
Initialize $G \leftarrow \{\}$
foreach $x \in \mathcal{S}$ **do**
 Neigh \leftarrow the set of vertices in $D_1(\mathcal{S})$ adjacent to x
 closest $\leftarrow \arg \min_{y \in \text{Neigh}} \{d(x, y)\}$
 CompNeigh $\leftarrow \{y \in \text{Neigh} \mid (\text{closest}, x, y) \text{ is compatible}\}$
 clComp $\leftarrow \arg \min_{y \in \text{CompNeigh}} \{d(x, y)\}$
 $G \leftarrow G \cup \{x, \text{closest}\}, \{x, \text{clComp}\}$
return G

Computing a Delaunay triangulation in the plane can be done in $O(n \log n)$ [11, Theorem 9.12]. The total number of edges in $D_1(\mathcal{S})$ is $O(n)$, so the sum of the sizes of all the Neigh over all $x \in \mathcal{S}$ is $O(n)$. Thus, the total running time of the for-loop is $O(n)$. This gives a running time for COMPATIBLE-CRUST of $O(n \log n + n) = O(n \log n)$ for $d = 2$. For $d \geq 3$, the Delaunay triangulation may have a size as large as $\Theta(n^{\lceil d/2 \rceil})$ [19, Chapter 27.1], in which case COMPATIBLE-CRUST does not do better than NN-COMPATIBLE for $d \in \{3, 4\}$ and does worse for $d \geq 5$.

It remains to be proved that the algorithms output $G_{\mathcal{C}}(\mathcal{S})$. Since COMPATIBLE-CRUST restricts itself to the set of edges of the Delaunay triangulation, we need to know that this set contains the edges of $G_{\mathcal{C}}(\mathcal{S})$. In the planar case, this is proved in [3, Lemma 11]. We extend the result to higher ambient dimensions in Corollary 5.

Finally, we need to prove that the closest and “closest compatible” neighbors to a sample point are indeed the adjacent vertices in $G_{\mathcal{C}}(\mathcal{S})$. As the proof is rather long and technical, we devote a full section to it, which we split into three subsections: In Section 3.1, we prove a sequence of lemmas about the local behavior of \mathcal{S} and \mathcal{C} . Then, in Section 3.2, we prove lower bounds on the angle between certain triples of points on \mathcal{C} ; in particular, Lemma 11 implies that consecutive triples of points have to be compatible. Lastly, in Section 3.3, we use the results from the first two subsections to prove that the edges constructed by the algorithms are indeed exactly the edges in $G_{\mathcal{C}}(\mathcal{S})$. Some of the proofs are omitted and appear only in the appendix of the full version of the paper [8].

3.1 Basic observations about \mathcal{S} and \mathcal{C}

Recall that \mathcal{S} is assumed to be an ϵ -sample of \mathcal{C} . In this subsection, we assume $\epsilon \leq 1$. Later, we will restrict ϵ to smaller values and state our assumptions on ϵ explicitly in each case.

For $p \in \mathcal{C}$, define $d_p = d(p, \mathcal{S})$. By definition of cl and ϵ -sample, cl is defined in $B_p\left(\frac{d_p}{\epsilon}\right)$. Since we assume $\epsilon \leq 1$, cl is in particular defined in $B_p(d_p)$. We will use the following lemma throughout the paper without referring to it explicitly.

► **Lemma 2.** *cl is continuous.*

Proof. Let $x \in \mathbb{R}^d \setminus \mathcal{M}$, and let x_1, x_2, \dots be a sequence of points in $\mathbb{R}^d \setminus \mathcal{M}$ that converges to x . To show that cl is continuous, it is enough to show that the image of the sequence under cl converges to $\text{cl}(x)$. Let y be an accumulation point in \mathcal{C} of the sequence $\text{cl}(x_1), \text{cl}(x_2), \dots$, which exists by compactness of \mathcal{C} . Then $d(x, y) \leq d(x, y')$ for any $y' \in \mathcal{C}$, so $y = \text{cl}(x)$. Thus, $\text{cl}(x)$ is the only accumulation point of $\text{cl}(x_1), \text{cl}(x_2), \dots$, so by compactness of \mathcal{C} , the sequence converges to $\text{cl}(x)$. ◀

► **Lemma 3.** *Let $x \in \mathbb{R}^d$ and $q \in \mathcal{C}$ be such that \overline{xq} does not intersect the medial axis. Let $p = \text{cl}(x)$. Then the interior of $B_x(d(x, q))$ contains either $[p, q]$ or $(q, p]$.*

Proof. By continuity of cl and connectedness of \overline{xq} , $\text{cl}(\overline{xq})$ must contain either $[p, q]$ or $[q, p]$. Suppose the former. Then for any $z \in [p, q]$, $z = \text{cl}(i)$ for some $i \in \overline{xq}$. Thus,

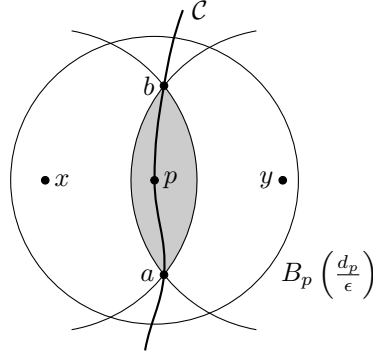
$$d(x, q) = d(x, i) + d(i, q) > d(x, i) + d(i, z) \geq d(x, z).$$

The statement follows, and the argument for $[q, p]$ is exactly the same. ◀

► **Lemma 4.** *Let $a \rightarrow b$ and $p \in (a, b)$. Then $d_p = \min\{d(p, a), d(p, b)\}$, and $d_p < d(p, s)$ for all $s \in \mathcal{S} \setminus \{a, b\}$.*

Proof. Suppose $s \notin \{a, b\}$ is a point in \mathcal{S} minimizing the distance to p , so $d_p = d(p, s)$. Then $B_p(d(p, s)) = B_p(d_p)$ and thus \overline{ps} does not intersect the medial axis. Since $\text{cl}(p) = p$, Lemma 3 (with $x = p$ and $q = s$) shows that the interior of $B_p(d_p)$ contains either a or b , which is a contradiction, as then either $d(p, a)$ or $d(p, b)$ would be smaller than $d(p, s)$. Thus, d_p is equal to either $d(p, a)$ or $d(p, b)$. ◀

As a step in proving the correctness of COMPATIBLE-CRUST, we need to show that for $a \rightarrow b$, there is an edge between a and b in the Delaunay triangulation of \mathcal{S} . Since we do not assume that \mathcal{S} is in general position, we do not know that there is a unique Delaunay



■ **Figure 1** The planar case with $X(a, b) = \{x, y\}$. The shaded area is $U(a, b)$ and contains $[a, b]$ by Lemma 8 (iii). By Lemma 8 (ii), $B_p\left(\frac{d_p}{\epsilon}\right)$ contains $X(a, b)$, where p is the midpoint on $[a, b]$.

triangulation of \mathcal{S} . Still, we know that if there is a closed ball B such that $B \cap \mathcal{S} = \{a, b\}$, then any Delaunay triangulation of \mathcal{S} has an edge between a and b . In the special case of curves in the plane, the following was proved in [3, Lemma 11].

► **Corollary 5.** *Let $a \rightarrow b$. Then there is an edge between a and b in any Delaunay triangulation of \mathcal{S} .*

Proof. Let p be a midpoint on $[a, b]$. By Lemma 4, $B_p(d_p) \cap \mathcal{S} = \{a, b\}$, so there is an edge between a and b in the Delaunay triangulation of \mathcal{S} . ◀

For $x, y \in \mathbb{R}^d$, let $E(x, y)$ be the set of points in \mathbb{R}^d that are equidistant from x and y .

► **Lemma 6.** *Let $b \in \mathcal{S}$, let $a \neq b$ be in the same connected component of \mathcal{C} as b , let p be either the midpoint on $[a, b]$ or equal to a , and assume $d_p = d(p, b)$. Then for every $x \in B_p\left(\frac{d_p}{\epsilon}\right) \cap E(a, b)$,*

- (i) $cl(x) \in (a, b)$,
- (ii) $(a, b) \subset B_x(d(x, b))$.

Proof. (i): Let $B = B_p\left(\frac{d_p}{\epsilon}\right)$, and let m be the midpoint on $[a, b]$. If $p = a$, then by Lemma 3, $m \in B$. Trivially, $m \in B$ also holds if $p = m$. Since \mathcal{S} is an ϵ -sample, B does not intersect the medial axis, so $cl : B \rightarrow \mathcal{C}$ is well-defined. Clearly, $cl(m) = m$, and $a, b \notin cl(B \cap E(a, b))$, as $d(a, x) = d(b, x)$ for every $x \in E(a, b)$. Since cl is continuous and $B \cap E(a, b)$ connected, we get that $cl(B \cap E(a, b)) \subset (a, b)$.

(ii): Since $\overline{xb} \subset B_p\left(\frac{d_p}{\epsilon}\right)$, Lemma 3 tells us that $[cl(x), b]$ is in the interior of $B_x(d(x, b))$ (since $a \in (b, cl(x)]$ is not in the interior of $B_x(d(x, b)) = B_x(d(x, a))$), and so must $(a, cl(x)]$ by a symmetric argument. ◀

► **Definition 7.** *For $a \neq b \in \mathbb{R}^d$, let $X(a, b)$ be the set of x such that $d(x, a) = d(x, b) = \frac{d(a, b)}{\epsilon\sqrt{4-\epsilon^2}}$, and let $U(a, b) = \bigcap_{x \in X(a, b)} B_x\left(\frac{d(a, b)}{\epsilon\sqrt{4-\epsilon^2}}\right)$, which is equal to $\bigcap_{x \in X(a, b)} B_x(d(x, a))$.*

► **Lemma 8.** *Let $a \rightarrow b$.*

- (i) *Let $p' \in E(a, b) \cap \partial U(a, b)$, and let x be the point in $X(a, b)$ maximizing the distance to p' . Then $d(p', a) = \epsilon d(p', x)$, $d(p', x) = d(a, x)$ and $2\angle axp' = \angle axb$.*
- (ii) *Let p be the midpoint of $[a, b]$. Then $X(a, b) \subset B_p\left(\frac{d_p}{\epsilon}\right)$.*
- (iii) $(a, b) \subset U(a, b)$.

See Figure 1 for an illustration of (ii) and (iii). We prove the lemma in [8, Appendix A.1].

3.2 Restrictions of angles between points on \mathcal{C}

With help from the results of the previous subsection, we now prove results that essentially limit the curvature of \mathcal{C} locally.

► **Proposition 9.** *Let $\epsilon \leq 0.765$, and let $a \rightarrow b \rightarrow c$ with $p \in (a, b)$ and $d(p, b) \leq d(p, a)$. Then for any x such that $d(x, p) = d(x, b) = \frac{d_p}{\epsilon}$, $(b, c] \cap B_x\left(\frac{d_p}{\epsilon}\right) = \emptyset$.*

The rough idea of the proof is to assume there is a $c' \in (b, c] \cap B_x\left(\frac{d_p}{\epsilon}\right)$ and consider a line segment \overline{xm} , where x satisfies the conditions in the lemma and m is the midpoint on $\overline{bc'}$. One can show that cl is defined on \overline{xm} , that $\text{cl}(x) \in (p, b)$, and that $\text{cl}(m) \in (b, c')$ and derive that $\text{cl}(\overline{xm})$ is disconnected, which is a contradiction by continuity of cl . We give the full details in [8, Appendix A.2].

► **Corollary 10.** *Let $\epsilon \leq 0.66$, let $a \rightarrow b \rightarrow c$, and let p be the midpoint of $[a, b]$ and $q \in (b, c]$. Then*

$$\angle pbq > 70.73^\circ + \arccos\left(0.33 \frac{d(q, b)}{d(p, b)}\right).$$

In particular, if $d(p, b) \geq d(q, b)$, then $\angle pbq > 141^\circ$.

Proof. We restrict our attention to a plane containing p, b and q and assume without loss of generality that $p = (0, -1)$, $b = (0, 0)$ and that q is not to the left of the y -axis. By Proposition 9, q cannot be in the disc D with radius $\frac{1}{\epsilon}$ with p and b on the boundary and center x to the right of the y -axis. Under this condition, we have $\angle pbq > \angle pbq'$, where q' is on the boundary of D above the x -axis and $d(q', b) = d(q, b)$. As illustrated in Figure 2a, $\cos \angle pbx = \frac{1/2}{1/\epsilon} \leq 0.33$. Similarly, $\cos \angle x bq' = \frac{d(q', b)/2}{1/\epsilon} \leq 0.33d(q', b)$. Since \arccos is decreasing, we get

$$\begin{aligned} \angle pbq &> \angle pbq' \\ &= \angle pbx + \angle x bq' \\ &\geq \arccos(0.33) + \arccos(0.33d(q', b)) \\ &> 70.73^\circ + \arccos(0.33d(q', b)). \end{aligned}$$

If we do not assume $d(p, b) = 1$, we have to replace $d(q', b)$ with $\frac{d(q', b)}{d(p, b)}$ in the last expression. Since $d(q', b) = d(q, b)$, this yields the wanted inequality. If $d(p, b) \geq d(q, b)$, then this lower bound is weakest when $d(q, b) = d(p, b)$. In this case the right-hand side is $> 141.46^\circ$. ◀

► **Lemma 11.** *Let $\epsilon \leq 0.765$, and let $a \rightarrow b \rightarrow c$. Then $(b, c] \cap B_x(d(x, a)) = \emptyset$ for all $x \in X(a, b)$.*

This proof is similar to that of Proposition 9; see [8, Appendix A.3] for the details.

► **Definition 12.** *We call a triple (a, b, c) of sample points compatible if $c \notin B_x(d(x, b))$ for all $x \in X(a, b)$ and $a \notin B_y(d(y, b))$ for all $y \in X(b, c)$.*

See Figure 3. Lemma 11 then implies that if $a \rightarrow b \rightarrow c$, then (a, b, c) is compatible.

► **Lemma 13.** *Let $\epsilon \leq 0.66$ and suppose (a, b, c) is compatible. Then*

$$\angle abc > 51.45^\circ + \arccos\left(0.6231 \frac{d(c, b)}{d(a, b)}\right).$$

In particular, $\angle abc > 102.9^\circ$.

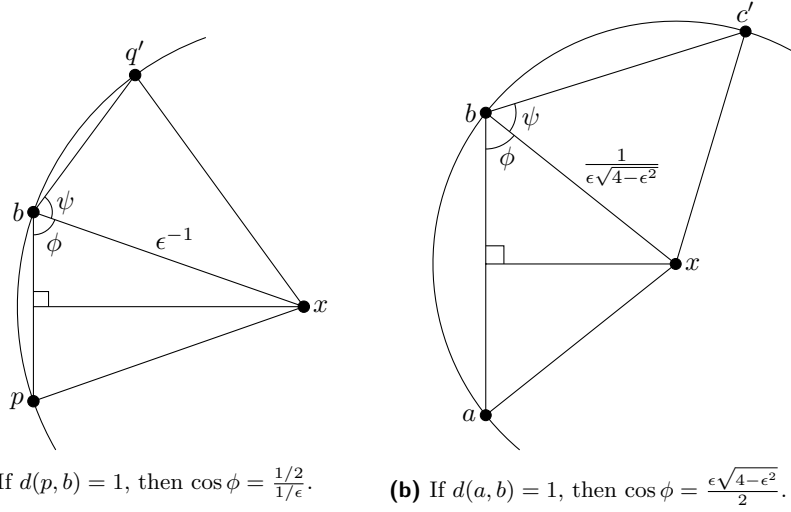


Figure 2

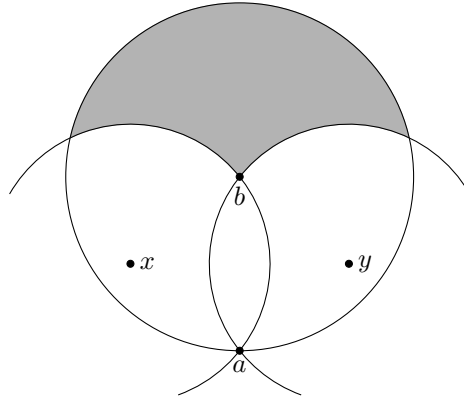
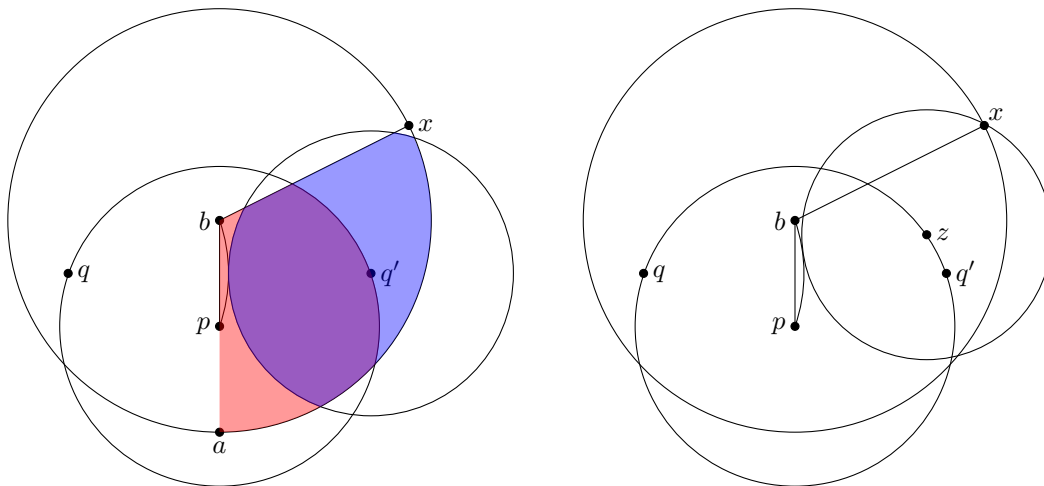


Figure 3 The planar case with $X(a, b) = \{x, y\}$. If $d(b, c) \leq d(a, b)$, then (a, b, c) is compatible if and only if c is in the shaded area.

Proof. We use an argument very similar to that in the proof of Corollary 10. We restrict our attention to the plane spanned by a, b and c and assume without loss of generality that $a = (0, -1), b = (0, 0)$ and that c is not to the left of the y -axis. By definition of compatibility, $c \notin B_x(d(x, a))$, where x is the element of $X(a, b)$ to the right of the y -axis. Under this condition, we have $\angle abc > \angle abc'$, where c' satisfies $d(c, b) = d(c', b)$ and is on the boundary of $B_x(d(x, a))$ above the x -axis. By definition of $X(a, b)$, we have $\cos \angle abx = \frac{\epsilon\sqrt{4-\epsilon^2}}{2}$, as illustrated in Figure 2b. Similar considerations show that $\cos \angle xbc' = \frac{\epsilon\sqrt{4-\epsilon^2}d(c', b)}{2d(a, b)}$. We have $d(c', b) = d(c, b)$ by assumption, and $\frac{\epsilon\sqrt{4-\epsilon^2}}{2} \leq \frac{0.66\sqrt{4-0.66^2}}{2} < 0.6231$. Since arccos is decreasing, this yields

$$\begin{aligned} \angle abc &> \angle abc' \\ &= \angle abx + \angle xbc' \\ &> \arccos(0.6231) + \arccos\left(0.6231 \frac{d(c, b)}{d(a, b)}\right) \end{aligned}$$

and then the wanted inequality follows from $\arccos(0.6231) > 51.45^\circ$.



(a) The discs $B_p\left(\frac{d(p,b)}{\epsilon}\right)$ (red) and $B_{q'}(d(q,q') - \epsilon^{-1})$ (blue) cover the relevant area except a small part close to x . (b) The distance from z to $B_q(\epsilon^{-1})$ is slightly smaller than $d(z,x)$.

■ Figure 4

(a, b, c) is compatible if and only if (c, b, a) is, so the inequality holds also if we switch a and b . Thus, we can assume $d(a, b) \geq d(c, b)$. Under this assumption, the right-hand side is smallest when $d(c, b) = d(a, b)$. Thus,

$$\angle abc > 2 \arccos(0.6231) > 102.9^\circ. \quad \blacktriangleleft$$

3.3 The closest compatible neighbors are the correct neighbors

In the runtime analysis of our algorithms, we stated that checking if a triple (a, b, c) of points is compatible can be done in constant time. Since we only need to consider the geometry of three fixed points, this is clear; to be precise, by arguments similar to those in the proof of Lemma 13, what we need to check is if

$$\angle abc > \arccos\left(\frac{0.66\sqrt{4 - 0.66^2}}{2}\right) + \arccos\left(\frac{0.66\sqrt{4 - 0.66^2}d(c, b)}{2d(a, b)}\right)$$

and the same with a and c switching places.

Recall that our algorithms construct edges from $b \in \mathcal{S}$ to a and c , where a is the closest point in \mathcal{S} to b , and c is the closest point in \mathcal{S} to b such that (a, b, c) is compatible. (COMPATIBLE-CRUST is restricted to the Delaunay neighbors, which by Corollary 5 is not a problem.) Since b has exactly two adjacent vertices in $G_{\mathcal{C}}(\mathcal{S})$, it is sufficient to prove that a, b and c are consecutive. This is exactly the statement of Proposition 16 below, which therefore finishes the proof of Theorem 1.

► **Lemma 14.** *Let $\epsilon \leq 0.66$ and $a \rightarrow b$, let p be the midpoint on $[a, b]$, and let c be a point on $\mathcal{C} \setminus [a, b]$ with $d(b, c) \leq d(a, b)$. Then $\angle pbc > 117.3^\circ$.*

Proof. Assume $\angle pbc \leq 117.3^\circ$, and let us restrict ourselves to a plane containing p, b, c . Without loss of generality, we can assume that $p = (0, -1)$, $b = (0, 0)$, and that c is not to the left of the y -axis. Let q be the point to the left of the y -axis such that $d(q, b) = d(q, p) =$

9:10 Tighter Bounds for Reconstruction from ϵ -Samples

ϵ^{-1} , and let q' be the reflection of q across the y -axis. By Lemma 6 (i), $\text{cl}(q') \in (p, b)$, and by Lemma 6 (ii) (choose $a = p$ in the lemma), $(p, b) \subset B_q(\epsilon^{-1})$. It follows that $c \notin B_{q'}(d(q, q') - \epsilon^{-1})$.

We have two remaining possibilities under the assumptions $\angle pbc \leq 117.3^\circ$ and $d(b, c) \leq d(a, b)$:

(i) $c \in B_b(d(a, b)) \cap B_p\left(\frac{d(p, b)}{\epsilon}\right)$,

(ii) $c \in B_b(d(a, b)) \setminus \left(B_p\left(\frac{d(p, b)}{\epsilon}\right) \cup B_{q'}(d(q, q') - \epsilon^{-1})\right)$,

To show that (i) is impossible, first assume that c is below or on the line l through q and q' . If \overline{cq} does not intersect \overline{ab} , let $I = \overline{cq}$. Otherwise, let $I = \overline{cq'}$. c is closer to any point on I than a is, so $a \notin \text{cl}(I)$. Since no point on I is above l , $b \notin \text{cl}(I)$, as p is always at least as close as b . But clearly, $\text{cl}(c) = c$, and we have already observed that $\text{cl}(q') \in (p, b)$, and $\text{cl}(q) \in (p, b)$ holds for the same reason. Thus, $\text{cl}(I)$ is disconnected, which contradicts the continuity of cl .

If instead c is above l , let $I = \overline{cq'}$ and use a similar argument with a and b exchanged.

Finally, we assume (ii), which is the case that requires the most care. Let $z = (1.244, -0.1351)$. Some calculation shows that $z \in B_p(\epsilon^{-1}) = B_p\left(\frac{d_p}{\epsilon}\right)$. Let $I = \overline{zq'}$. Since $I \subset B_p\left(\frac{d_p}{\epsilon}\right)$, I does not intersect the medial axis of C .

Let x be the intersection of the ray from b into the first quadrant with angle $\angle 117.3^\circ$ with the boundary of $B_b(2)$. As Figure 4a illustrates, c must be in an area close to x , and x is the point in this area furthest away from z . Some more calculation shows that $d(z, x) < 1.18 < d(z, q) - \epsilon^{-1}$; see Figure 4b. This means that z is closer to c than to any point on $[p, b]$, since $[p, b] \subset B_q(\epsilon^{-1})$, as we have observed. Thus, $\text{cl}(z) \notin [p, b]$. In addition, $\text{cl}(q') \in (p, b)$ by Lemma 6 (i). But all points on I are closer to c than to both p and b (it is enough to check the endpoints of I), so $p, b \notin \text{cl}(I)$. Thus, $\text{cl}(I)$ is disconnected, which is impossible, as cl is continuous. \blacktriangleleft

► **Proposition 15.** *Let $\epsilon \leq 0.66$, and let a be a sample point and b a closest neighbor to a among the other sample points. Then a and b are consecutive.*

Proof. Suppose for a contradiction that x , a and y are consecutive and $b \notin \{x, y\}$. Let p be the midpoint of $[x, a]$ and q the midpoint of $[a, y]$. By Corollary 10, $\angle paq > 141^\circ$, and by Lemma 14, both $\angle pab$ and $\angle qab$ are greater than 117.3° , as $d(x, a), d(y, a) \geq d(a, b)$. The sum of these angles is greater than 360° , which is impossible. \blacktriangleleft

► **Proposition 16.** *Let $\epsilon \leq 0.66$. Let b be a sample point, a a closest sample point to b , and c the closest sample point to b such that (a, b, c) is compatible. Then a , b and c are consecutive.*

In particular, there is a unique closest point c to a such that (a, b, c) is compatible.

The idea of the proof is as follows: We let a , b and c be as in the proposition, suppose there is a $c' \neq c$ such that a , b and c' are consecutive, let q be the midpoint of $[b, c']$, and carefully pick a point $p \in [a, b]$. We get lower bounds on $\angle qbc$ and $\angle pbq$ by Lemma 14 and Proposition 9 depending on the distances from q , c and p to b . This gives an upper bound on $\angle cbp$, which leads to a contradiction by an argument similar to the one in the proof of Lemma 14. However, the proof is complicated by the degrees of freedom we have in choosing the distances from the various points to b . We give the details in [8, Appendix A.4].

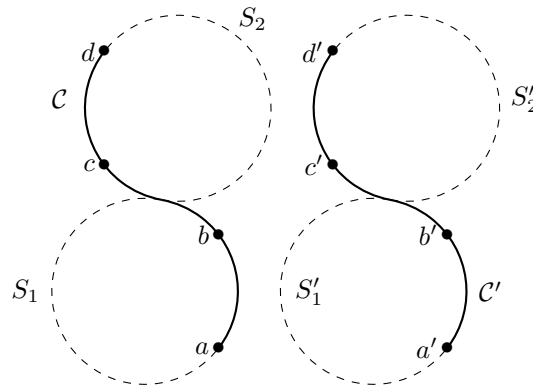


Figure 5 The first step in the construction of the curves of Theorem 17.

4 Counterexample to curve reconstruction for $\epsilon = 0.72$

In this section, we prove the following theorem, which says that correct curve reconstruction using 0.72-samples is not in general possible, even in \mathbb{R}^2 . Moreover, one cannot determine whether the (union of) curve(s) has more than one connected component, and the reconstruction problem remains impossible also under the assumption that the sample is taken from a single connected curve.

► **Theorem 17.** *There is a finite set $S \subset \mathbb{R}^2$ that is a 0.72-sample of C_1, C_2, C_3 and C_4 , where C_1 and C_2 are connected closed curves and C_3 and C_4 are disconnected unions of closed curves, and $G_{C_i}(S) \neq G_{C_j}(S)$ for all $i \neq j$.*

As we will construct subsets of the curves before we construct the complete curves, we extend the definition of ϵ -sampling to unions of closed curves in the obvious way.

Let $a = (0, -1), b = (0, 0), c = (-1.008, 0.614), d = (-1.008, 1.614)$. Let S_1 and S_2 be the two tangent circles with the same radius such that $a, b \in S_1, c, d \in S_2$ and the tangent point is the midpoint q between b and c . Let C be the union of the part of S_1 running from a to q through b and the part of S_2 running from q to d through c .

Next, let a', b', c', d' be the points, S'_1, S'_2 the circles and C' the curve we get by translating the whole construction horizontally to the right so that $d(b, c) = d(b, c')$; see Figure 5. Let T be the set of midpoints of $[a, b], [b, c], [c, d], [a', b'], [b', c']$ and $[c', d']$.

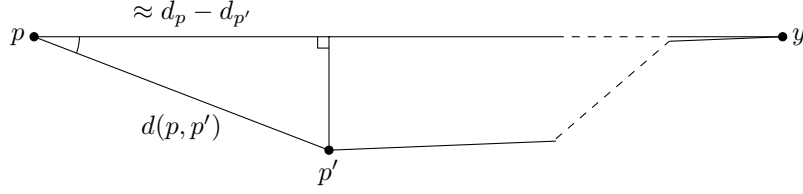
► **Lemma 18.** *If $\{a, b, c, d, a', b', c', d'\}$ is not a 0.72-sample of $C \cup C'$, then there is a $t \in T$ such that $B_t(\frac{d_t}{0.72})$ intersects the medial axis of $C \cup C'$.*

Proof. By definition, if $\{a, b, c, d, a', b', c', d'\}$ is not a 0.72-sample of $C \cup C'$, then there is a $p \in C \cup C'$ such that $B_p(\frac{d_p}{0.72})$ intersects the medial axis of $C \cup C'$. Thus, it is enough to show that for every $p \in C \cup C' \setminus T$, there is a $t \in T$ such that $B_p(\frac{d_p}{0.72}) \subset B_t(\frac{d_t}{0.72})$.

Let $p \in (x, y) \subset C$ for some $x \rightarrow y$. We know that $d_p = d(p, x)$ or $d_p = d(p, y)$ by Lemma 4. Suppose $d_p = d(p, y)$ ($d_p = d(p, x)$ is similar), and pick $p' \in (p, y)$. Let $B = B_p(\frac{d_p}{0.72})$ and $B' = B_{p'}(\frac{d_{p'}}{0.72})$. If $B \not\subset B'$, there is a point on the ray from p through p' in $B' \setminus B$, which means that $\frac{d_p}{0.72} < d(p, p') + \frac{d_{p'}}{0.72}$, or equivalently

$$\frac{d_p - d_{p'}}{d(p, p')} < 0.72.$$

9:12 Tighter Bounds for Reconstruction from ϵ -Samples



■ **Figure 6** Assuming $d(p, p') \ll d(p, y)$, we have $\cos(\angle p'py) \approx \frac{d_p - d_{p'}}{d(p, p')}$. The dotted lines represent that we have collapsed a large part of the figure.

Observe that if we let p' approach p , then $\frac{d_p - d_{p'}}{d(p, p')}$ approaches $\cos \angle p'py$; see Figure 6. One can check that $\angle p'py < 40^\circ$ for the possible p and p' in our example (by a large margin), while $\arccos(0.72) > 43^\circ$. Thus,

$$\arccos(0.72) > \arccos\left(\frac{d_p - d_{p'}}{d(p, p')}\right)$$

for p' sufficiently close to p , so $0.72 < \frac{d_p - d_{p'}}{d(p, p')}$, a contradiction. This proves that as p moves along \mathcal{C} or \mathcal{C}' from a point in T towards a point in $\{a, b, c, d\}$, the disc $B_p\left(\frac{d_p}{0.72}\right)$ decreases (in the sense that later discs are contained in earlier discs), proving the lemma. ◀

► **Lemma 19.** $\{a, b, c, d, a', b', c', d'\}$ is a 0.72-sample of $\mathcal{C} \cup \mathcal{C}'$.

Proof. By Lemma 18, what we need to show is that for any $t \in T$, $B_t\left(\frac{d_t}{0.72}\right)$ does not intersect the medial axis.

To reduce the problem, observe that we have symmetry around the midpoint between b and c' , as $\vec{bc} = -\vec{c'b}$ and $\vec{cd} = -\vec{d'a}$. Thus, we can restrict ourselves to the midpoints p, q and r of $[a, b]$, $[b, c]$ and $[c, d]$, respectively. For the rest of the proof, assume $t \in \{p, q, r\}$.

We extend cl to a set-valued map from \mathbb{R}^2 to $\mathcal{C} \cup \mathcal{C}'$ by letting $\text{cl}(p)$ be the set of points in $\mathcal{C} \cup \mathcal{C}'$ that minimize the distance to p . Let m be a point such that $\text{cl}(m)$ contains at least two points in $\mathcal{C} \cup \mathcal{C}'$, and let x and y be distinct points in $\text{cl}(m)$. We will show that for $t \in \{p, q, r\}$, $m \notin B_t\left(\frac{d_t}{0.72}\right)$.

There are the following cases to consider:

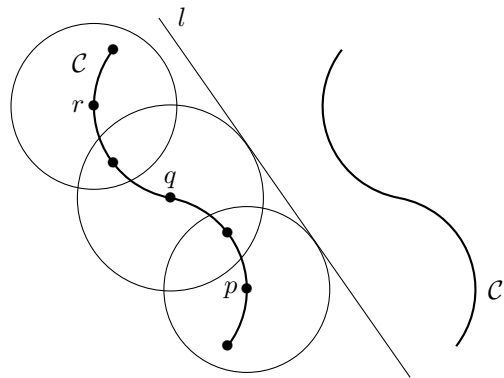
- $x, y \in \mathcal{C}$,
- $x, y \in \mathcal{C}'$,
- $x \in \mathcal{C}$ and $y \in \mathcal{C}'$.

In the first case, m is on the medial axis of \mathcal{C} . This has two connected components: one is a curve starting at the center s_1 of S_1 and going leftwards and downwards from there, and the other is the mirror image through q of the first one. Because of symmetry, we only have to consider the first component. On this curve, s_1 minimizes the distance to p and q , and r is far away from the whole curve. One can check that the radius of S_1 is greater than 0.82, that $d_q = d(q, b) < 0.59$ and that $d_q > d_p$. Thus,

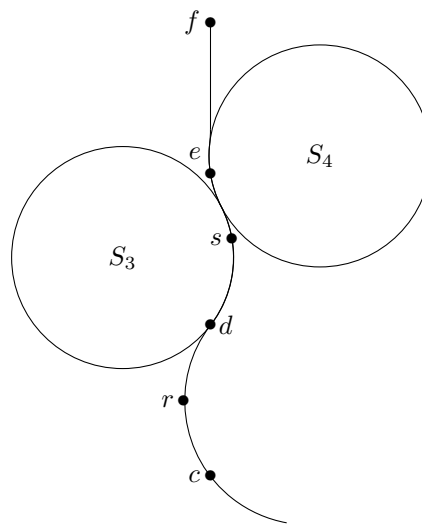
$$d(s_1, p) = d(s_1, q) > 0.82 > \frac{d_q}{0.72} > \frac{d_p}{0.72},$$

so we conclude that $B_t\left(\frac{d_t}{0.72}\right)$ does not intersect the medial axis of \mathcal{C} .

Next, we assume that $x, y \in \mathcal{C}'$. Then there is a point m' on the line segment \overline{mt} such that $\text{cl}(m')$ intersects both \mathcal{C} and \mathcal{C}' , so m' is on the medial axis. If $m \in B_t\left(\frac{d_t}{0.72}\right)$, then $m' \in B_t\left(\frac{d_t}{0.72}\right)$, so we have reduced the second case to the third case.



■ **Figure 7** Illustration for the proof of Lemma 19.

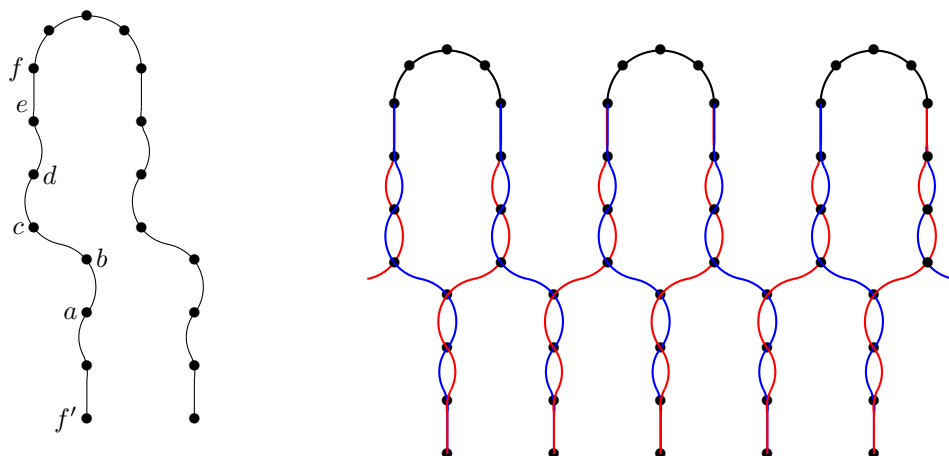


■ **Figure 8** \mathcal{C} is extended from d through e to f , where the tangent is vertical.

Lastly, assume that $x \in \mathcal{C}$ and $y \in \mathcal{C}'$; see Figure 7. Let l be the perpendicular bisector of s_1 and the center s'_2 of S'_2 . If $y \in S'_2$, then m is either on l or to the right of l (the latter can only happen if $x \in S_2$). By numerical calculation, one can check that $l \cap B_t(\frac{d_t}{0.72}) = \emptyset$, so in this case, $m \notin B_t(\frac{d_t}{0.72})$. At the same time, if $y \in S'_1$, then $d(t, y) > \frac{2d_t}{0.72}$, so if $m \in B_t(\frac{d_t}{0.72})$, then $y \notin S'_1$, as t is closer to m than S'_1 is. ◀

Now we want to extend this construction. See Figure 8 for what follows. We add a point e such that d is the midpoint between c and e . Next, we put a circle S_3 with radius $\frac{d_r}{0.72}$ so that it is tangent to S_2 at d , and a circle S_4 with the same radius as S_3 tangent to S_3 such that e lies on S_4 . If we extend \mathcal{C} such that it contains $[d, e]$ along S_3 and S_4 in the obvious way, then $\{a, b, c, d, e\}$ is a 0.72-sample of \mathcal{C} . To see this, note that if s is the midpoint of $[d, e]$, then the difference in x -coordinate between s and d is less than that between r and d , so $d_s < d_r$. The closest points on the medial axis to s are the centers of S_3 and S_4 , which have a distance of $\frac{d_r}{0.72} > \frac{d_s}{0.72}$ to s .

The tangent of \mathcal{C} at d is much closer to being vertical than the tangent at e , and if we add another point f such that e is the midpoint between d and f , then we can extend \mathcal{C} to f similarly to how we extended \mathcal{C} from d to e in such a way that $\{a, b, c, d, e, f\}$ is a 0.72-sample, and such that the tangent of \mathcal{C} at f is vertical.



(a) \mathcal{C} and \mathcal{C}' are extended and tied together. At f and f' , the tangents of the curve are vertical.

(b) Together with the black semicircles, the blue and red curves both give a valid reconstruction under the 0.72-sampling condition.

■ Figure 9

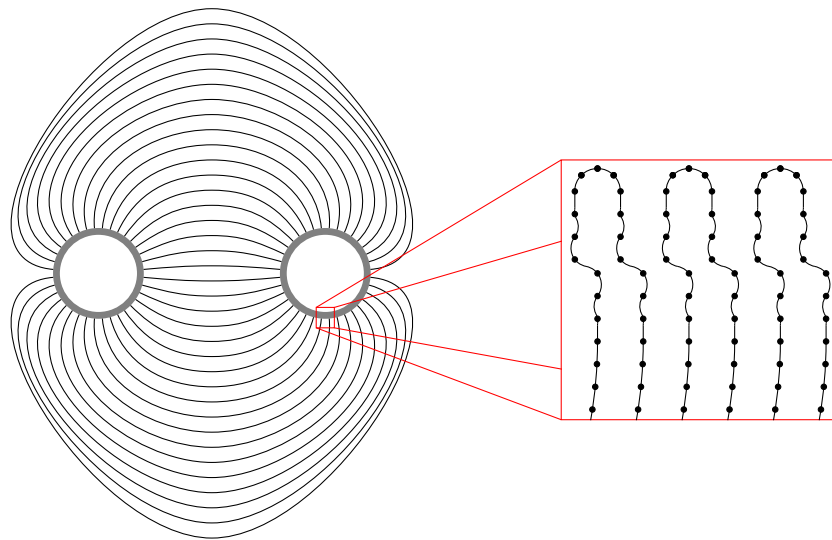
We can do the same below a , adding two points such that \mathcal{C} can be extended downwards and the tangent of \mathcal{C} at the lowest point is vertical. Now do the same for \mathcal{C}' , and add a sequence of points densely sampling a semicircle to connect \mathcal{C} and \mathcal{C}' as shown in Figure 9a. Again, the points shown make up a 0.72-sample of the curve. Next, we put many copies of this construction next to each other as shown in Figure 9b. Each copy is translated horizontally such that $d(b, c')$ is equal to the distance between b' in one copy and c in the copy on its right. If we ignore what happens to the far right or left, there are two ways to draw a set of curves with endpoints among the bottom points such that the set of points is a 0.72-sample of the union of curves.

We now take this long strip of points and curves and bend it slightly upwards such that they are contained in an annulus and the ends meet; see Figure 10. As the length of this strip goes to infinity, the distances from points on the curve to the closest sample point and the medial axis are distorted by a factor that approaches 1 when we bend it into the annulus. Our arguments for the the set of points being a 0.72-sample works equally well for an $\epsilon > 0.72$ sufficiently close to 0.72, so after turning the (sufficiently long) strip into an annulus, the point set stays a δ -sample for some $\delta > 0.72$.

Finally, we consider two such annuli with “the ends tied together”, meaning that we draw curves between endpoints in the first annulus and endpoints in the second annulus, and sample the curves densely; see Figure 10. In each of the two annuli, we have two choices of how to draw the curve, as illustrated in Figure 9b, which gives four different choices. Exactly two of these choices result in a connected curve, and in all four cases, the set of points is a 0.72-sample of the curve or union of curves. Summing up, we get Theorem 17.

5 Counterexample to hypersurface reconstruction for $\epsilon = 0.72$

We have not defined what “correct reconstruction” means in higher dimensions. But assuming that preserving the number of connected components is required, we show that correct reconstruction of hypersurfaces in \mathbb{R}^d using 0.72-samples is impossible for any $d \geq 2$.



■ **Figure 10** A reconstruction of the whole point set with the two annuli in grey. One can make sure that the curve is connected, and the point set is a 0.72-sample of it.

► **Theorem 20.** *For any $d \geq 2$, there is a finite point set $\mathcal{S} \subset \mathbb{R}^d$ that is a 0.72-sample of two manifolds \mathcal{C} and \mathcal{C}' without boundary of dimension $d - 1$ with a different number of connected components.*

Proof. The case $d = 2$ follows immediately from Theorem 17. For any point $p = (x, y) \in (0, \infty) \times \mathbb{R}$, let p° be the circle centered at $(0, y)$ containing p . For any set $X \subset (0, \infty) \times \mathbb{R}$, let $X^\circ = \bigcup_{p \in X} p^\circ$. Let \mathcal{C}_i be as in Theorem 17 for $1 \leq i \leq 4$, and let $\mathcal{S}_{\text{curve}}$ be the 0.72-sample as constructed in the previous section. Pick a constant R and translate \mathcal{C}_i so that it is contained in $(R, \infty) \times \mathbb{R}$. Similarly to how we bent a strip into a large annulus earlier, by choosing R large, we can make sure that a sufficiently dense subset \mathcal{S} of $\mathcal{S}_{\text{curve}}^\circ$ is a δ -sample of \mathcal{C}_i for some $\delta > 0.72$. Choosing $i = 1$ and $i = 3$, the theorem for $d = 3$ follows. To get the theorem for larger d , one can iterate the construction we used to get from $d = 2$ to $d = 3$. ◀

6 Discussion

We have only considered unions of closed curves. An obvious question is if our work generalizes to open curves. We expect that this can be dealt with by a slight tweak of the algorithms when the endpoints are far apart: Instead of immediately connecting a point to its “correct” neighbors (i.e., its closest and closest “compatible” neighbors), one should add an edge between two points only when both points consider the other as a “correct” neighbor. However, we have not tried to turn this intuition into a precise statement.

Though this paper is mainly about curve reconstruction, we hope that it can also be a step towards improving the sampling conditions for surface reconstruction. Our arguments are valid for samples in any ambient dimension, and we expect many of our intermediate results to carry over to points on surfaces instead of curves. We consider generalizing our approach to surface reconstruction to be a promising direction of future research.

References

- 1 Ahmed Abdelkader, Chandrajit L. Bajaj, Mohamed S. Ebeida, Ahmed H. Mahmoud, Scott A. Mitchell, John D. Owens, and Ahmad A. Rushdi. Sampling Conditions for Conforming Voronoi Meshing by the VoroCrust Algorithm. In *34th International Symposium on Computational Geometry (SoCG 2018)*, pages 1:1–1:16, 2018.
- 2 Ernst Althaus and Kurt Mehlhorn. Traveling salesman-based curve reconstruction in polynomial time. *SIAM Journal on Computing*, 31(1):27–66, 2001.
- 3 Nina Amenta, Marshall Bern, and David Eppstein. The crust and the β -skeleton: Combinatorial curve reconstruction. *Graphical models and image processing*, 60(2):125–135, 1998.
- 4 Nina Amenta, Sunghee Choi, Tamal K. Dey, and Naveen Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Proceedings of the sixteenth annual symposium on Computational geometry (SCG 2000)*, pages 213–222, 2000.
- 5 Dominique Attali. r -regular shape reconstruction from unorganized points. *Computational Geometry*, 10(4):239–247, 1998.
- 6 Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gael Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. A survey of surface reconstruction from point clouds. *Computer Graphics Forum*, 36(1):301–329, 2017.
- 7 Fausto Bernardini and Chandrajit L. Bajaj. Sampling and reconstructing manifolds using alpha-shapes. In *Proceedings of the 9th Canadian Conference on Computational Geometry (CCCG 1997)*, pages 193–198, 1997.
- 8 Håvard Bakke Bjerkevik. Tighter bounds for reconstruction from ϵ -samples. *arXiv preprint v2*, 2022. [arXiv:2112.03656](https://arxiv.org/abs/2112.03656).
- 9 Harry Blum. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.
- 10 Frédéric Chazal and André Lieutier. Smooth manifold reconstruction from noisy and non-uniform approximation with guarantees. *Computational Geometry*, 40(2):156–170, 2008.
- 11 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2008.
- 12 Luiz Henrique De Figueiredo and Jonas de Miranda Gomes. Computational morphology of curves. *The Visual Computer*, 11(2):105–112, 1994.
- 13 Tamal K. Dey. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*, volume 23. Cambridge University Press, 2006.
- 14 Tamal K. Dey, Joachim Giesen, Edgar A. Ramos, and Bardia Sadri. Critical points of distance to an ϵ -sampling of a surface and flow-complex-based surface reconstruction. *International Journal of Computational Geometry & Applications*, 18(1–2):29–61, 2008.
- 15 Tamal K. Dey and Piyush Kumar. A simple provable algorithm for curve reconstruction. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1999)*, volume 99, pages 893–894, 1999.
- 16 Tamal K. Dey, Kurt Mehlhorn, and Edgar A. Ramos. Curve reconstruction: Connecting dots with good reason. *Computational Geometry*, 15(4):229–244, 2000.
- 17 Tamal K. Dey and Rephael Wenger. Fast reconstruction of curves with sharp corners. *International Journal of Computational Geometry & Applications*, 12(05):353–400, 2002.
- 18 Christopher Gold. Crust and anti-crust: a one-step boundary and skeleton extraction algorithm. In *Proceedings of the fifteenth annual symposium on Computational geometry (SCG 1999)*, pages 189–196, 1999.
- 19 Jacob E. Goodman, Joseph O’Rourke, and Csaba D. Toth. *Handbook of Discrete and Computational Geometry*. CRC press, 3rd edition, 2017.
- 20 Tobias Lenz. How to sample and reconstruct curves with unusual features. In *EWCG: Proc. of the 22nd European Workshop on Computational Geometry*, pages 29–32. Citeseer, 2006.
- 21 Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

- 22 Stefan Ohrhallinger, Scott A. Mitchell, and Michael Wimmer. Curve reconstruction with many fewer samples. *Computer Graphics Forum*, 35(5):167–176, 2016.
- 23 Stefan Ohrhallinger, Jiju Peethambaran, Amal D. Parakkat, Tamal K. Dey, and Ramanathan Muthuganapathy. 2d points curve reconstruction survey and benchmark. *Computer Graphics Forum*, 40(2):611–632, 2021.
- 24 Peer Stalling. Topologically correct surface reconstruction using alpha shapes and relations to ball-pivoting. In *19th International Conference on Pattern Recognition (ICPR 2008)*, pages 1–4, 2008.
- 25 Peer Stalling and Leonid Tcherniavski. Provably correct reconstruction of surfaces from sparse noisy samples. *Pattern Recognition*, 42(8):1650–1659, 2009.

Erdős–Szekeres-Type Problems in the Real Projective Plane

Martin Balko  

Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Manfred Scheucher  

Institut für Mathematik, Technische Universität Berlin, Germany

Pavel Valtr 

Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Abstract

We consider point sets in the real projective plane \mathbb{RP}^2 and explore variants of classical extremal problems about planar point sets in this setting, with a main focus on Erdős–Szekeres-type problems.

We provide asymptotically tight bounds for a variant of the Erdős–Szekeres theorem about point sets in convex position in \mathbb{RP}^2 , which was initiated by Harborth and Möller in 1994. The notion of convex position in \mathbb{RP}^2 agrees with the definition of convex sets introduced by Steinitz in 1913.

For $k \geq 3$, an (*affine*) k -hole in a finite set $S \subseteq \mathbb{R}^2$ is a set of k points from S in convex position with no point of S in the interior of their convex hull. After introducing a new notion of k -holes for points sets from \mathbb{RP}^2 , called *projective k -holes*, we find arbitrarily large finite sets of points from \mathbb{RP}^2 with no projective 8-holes, providing an analogue of a classical result by Horton from 1983. We also prove that they contain only quadratically many projective k -holes for $k \leq 7$. On the other hand, we show that the number of k -holes can be substantially larger in \mathbb{RP}^2 than in \mathbb{R}^2 by constructing, for every $k \in \{3, \dots, 6\}$, sets of n points from $\mathbb{R}^2 \subset \mathbb{RP}^2$ with $\Omega(n^{3-3/5k})$ projective k -holes and only $O(n^2)$ affine k -holes. Last but not least, we prove several other results, for example about projective holes in random point sets in \mathbb{RP}^2 and about some algorithmic aspects.

The study of extremal problems about point sets in \mathbb{RP}^2 opens a new area of research, which we support by posing several open problems.

2012 ACM Subject Classification Theory of computation \rightarrow Randomness, geometry and discrete structures; Theory of computation \rightarrow Computational geometry; Mathematics of computing \rightarrow Combinatorics; Mathematics of computing \rightarrow Probability and statistics; Information systems \rightarrow Data structures

Keywords and phrases real projective plane, point set, convex position, k -gon, k -hole, Erdős–Szekeres theorem, Horton set, random point set

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.10

Related Version *Full Version:* <https://arxiv.org/abs/2203.07518>

Funding *Martin Balko:* supported by the grant no. 21-32817S of the Czech Science Foundation (GAČR), by the Center for Foundations of Modern Computer Science (Charles University project UNCE/SCI/004). This article is part of a project that has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 810115).

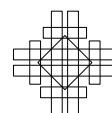
Manfred Scheucher: supported by the DFG Grant SCHE 2214/1-1.

Pavel Valtr: supported by the grant no. 21-32817S of the Czech Science Foundation (GAČR).



© Martin Balko, Manfred Scheucher, and Pavel Valtr;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 10; pp. 10:1–10:15
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

1.1 Erdős–Szekeres-type results in the Euclidean plane

Throughout the whole paper, we consider each set S of points from the Euclidean plane \mathbb{R}^2 to be finite and in *general position*, that is, no three points of S lie on a common line. We say that a set S of k points in the Euclidean plane is in *convex position* if S forms the vertex set of a convex polygon, which we call a *k-gon* or an *affine k-gon*.

In 1935, Erdős and Szekeres [16] showed that, for every integer $k \geq 3$, there is a smallest positive integer $ES(k)$ such that every finite set of at least $ES(k)$ points in the plane in general position contains a subset of k points in convex position. This result, known as the *Erdős–Szekeres theorem*, was one of the starting points of both discrete geometry and Ramsey theory. It motivated various lines of research that led to several important results as well as to many difficult open problems. For example, there were many efforts to determine the growth rate of the function $ES(k)$. Erdős and Szekeres [16] showed $ES(k) \leq \binom{2k-4}{k-2} + 1$ and conjectured that $ES(k) = 2^{k-2} + 1$ for every $k \geq 2$. This conjecture, known as the *Erdős–Szekeres conjecture*, was later supported by Erdős and Szekeres [17], who proved the matching lower bound $ES(k) \geq 2^{k-2} + 1$. The Erdős–Szekeres conjecture was verified for $k \leq 6$ [37] (see also [29, 33]), but is still open for $k \geq 7$. In fact, Erdős even offered \$500 reward for its solution. The currently best upper bound $ES(k) \leq 2^{k+O(\sqrt{k \log k})}$ is due to Holmsen, Mojarrad, Pach, and Tardos [25], who improved an earlier breakthrough by Suk [36] who showed $ES(k) \leq 2^{k+O(k^{2/3} \log k)}$. Altogether, these estimates give, for every $k \geq 2$,

$$2^{k-2} + 1 \leq ES(k) \leq 2^{k+O(\sqrt{k \log k})}. \quad (1)$$

Several variations of the Erdős–Szekeres theorem have been studied in the literature. In the 1970s, Erdős [15] asked whether there is a smallest positive integer $h(k)$ such that every set S of at least $h(k)$ points in the plane in general position contains an (*affine*) *k-hole*, which is a convex polygon spanned by a subset of k points from S that does not contain any point from S in its interior. In other words, a *k-hole* in a finite points set S in the plane in general position is a *k-gon* which is *empty* in S , that is, its interior does not contain any point from S . After Horton [26] constructed arbitrarily large point sets with no 7-hole, it took more than 20 years until Gerken [21] and Nicolas [31] independently showed that every sufficiently large set of points contains a 6-hole. Therefore, $h(k)$ is finite if and only if $k \leq 6$.

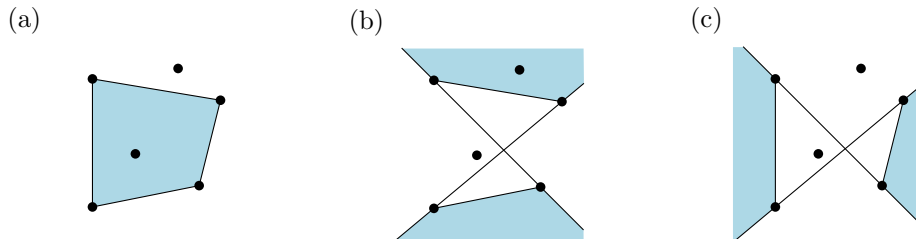
Estimating the minimum number of *k-holes* is another example of a classical Erdős–Szekeres-type problem. For a fixed integer $k \geq 3$ and a positive integer n , let $h_k(n)$ be the minimum number of *k-holes* in any finite set of n points in the plane. The growth rate of the function $h_k(n)$ was also studied extensively. Horton’s result implies $h_k(n) = 0$ for $k \geq 7$. The minimum numbers of 3- and 4-holes are known to be quadratic in n , but we only have the bounds $\Omega(n \log^{4/5} n) \leq h_5(n) \leq O(n^2)$ and $\Omega(n) \leq h_6(n) \leq O(n^2)$ [3, 9] for 5- and 6-holes, respectively. However, it is widely conjectured that h_5 and h_6 are also both quadratic in n .

In this paper, we consider analogous Erdős–Szekeres-type problems in the real projective plane \mathbb{RP}^2 . We define notions of convex position, *k-gons*, and *k-holes* in \mathbb{RP}^2 and study the corresponding extremal problems, providing several new results as well as numerous open problems in this new line of research.

1.2 Convex sets in the real projective plane

As in the planar case, we consider only sets P of points from the real projective plane \mathbb{RP}^2 that are finite and in *general position*, that is, no three points from P lie on a common projective line. We say that P is in *projective convex position* if it is a set in convex position

in some Euclidean plane $\rho \subset \mathbb{R}\mathcal{P}^2$. Recall that by removing a projective line from $\mathbb{R}\mathcal{P}^2$ one obtains a Euclidean plane. Following the notation introduced by Steinitz [35], we say that a subset X of $\mathbb{R}\mathcal{P}^2$ is *semiconvex* if any two points of X can be joined by a line segment fully contained in X . The set X is *convex* if it is semiconvex and does not contain some projective line, that is, X is contained in a plane $\rho \subset \mathbb{R}\mathcal{P}^2$; see also [13]. A *projective convex hull* of a set $Y \subset \mathbb{R}\mathcal{P}^2$ is an inclusion-wise minimal convex subset of $\mathbb{R}\mathcal{P}^2$ containing Y . We note that, unlike the situation in the plane, a projective convex hull of Y does not have to be determined uniquely; see Figure 1.



■ **Figure 1** An example of three projective 4-gons determined by the same subset of four points from a set P of six points in $\mathbb{R}\mathcal{P}^2$. The projective 4-gons in (a) and (b) are not projective 4-holes in P , but the projective 4-gon in (c) is a projective 4-hole in P .

► **Definition 1** (A projective k -gon). For a positive integer k and a finite set P of points from $\mathbb{R}\mathcal{P}^2$ in general position, a projective k -gon determined by P is a projective convex hull of a set I of k points from P which contains all points of I on its boundary; see Figure 1.

The notion “projective k -gon” in $\mathbb{R}\mathcal{P}^2$ is a natural analogue of the notion “affine k -gon” in \mathbb{R}^2 , since projective k -gons in $\mathbb{R}\mathcal{P}^2$ are exactly those subsets of $\mathbb{R}\mathcal{P}^2$ which are convex k -gons in some of the planes contained in $\mathbb{R}\mathcal{P}^2$.

Since a projective convex hull is not determined uniquely, a set of k points in $\mathbb{R}\mathcal{P}^2$ can determine several projective k -gons. In particular, it is not difficult to verify that

- (i) any three points in general position in $\mathbb{R}\mathcal{P}^2$ determine four projective 3-gons,
- (ii) any four points in general position in $\mathbb{R}\mathcal{P}^2$ determine three projective 4-gons,
- (iii) any five points in general position in $\mathbb{R}\mathcal{P}^2$ determine exactly one projective 5-gon, and
- (iv) any $k \geq 6$ points in general position in $\mathbb{R}\mathcal{P}^2$ determine at most one projective k -gon.

We also introduce the following natural analogue of holes in the real projective plane.

► **Definition 2** (A projective k -hole). For an integer $k \geq 3$ and a finite set P of points from $\mathbb{R}\mathcal{P}^2$ in general position, a projective k -hole in P is a projective k -gon determined by points from P that does not contain any point from P in its interior; see Figure 1.

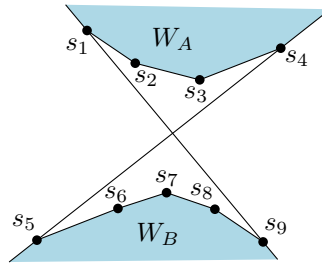
The notion of a “projective k -hole” in $\mathbb{R}\mathcal{P}^2$ is a natural analogue of the notion of an “(affine) k -hole” in \mathbb{R}^2 , since projective k -holes in $\mathbb{R}\mathcal{P}^2$ are exactly those subsets of $\mathbb{R}\mathcal{P}^2$ which are (affine) k -holes in some of the planes contained in $\mathbb{R}\mathcal{P}^2$.

We note that, again, a single set of $k \in \{3, 4\}$ points in general position in $\mathbb{R}\mathcal{P}^2$ can determine several different projective k -holes. Also note that, if H is a projective k -hole in a finite set P of points from $\mathbb{R}\mathcal{P}^2$ in general position, then in every affine plane $\rho \subset \mathbb{R}\mathcal{P}^2$ containing H , the set H is an affine k -hole. A subset of $\mathbb{R}\mathcal{P}^2$ is a *projective hole* in P if it is a projective k -hole in P for some integer $k \geq 3$.

We also describe the following alternative view on projective k -gons and k -holes via planar point sets. A *double chain* [27] is a set $S = A \cup B$ of k points from \mathbb{R}^2 with $A = \{s_1, \dots, s_m\}$ and $B = \{s_{m+1}, \dots, s_k\}$ for some m with $1 \leq m \leq k - 1$ such that, for every $i = 1, \dots, k$, the

line $\overline{s_i s_{i+1}}$ separates $A \setminus \{s_i, s_{i+1}\}$ from $B \setminus \{s_i, s_{i+1}\}$ (indices modulo k); see Figure 2. The sets A and B are the *chains* of the double chain. For a line ℓ not separating A , let H_ℓ^A be the closed half-plane bounded by ℓ that contains A and we similarly define H_ℓ^B . The *double chain k -wedge* of S is the union $W_A \cup W_B$ where $W_A = \bigcap_{i=0}^m H_{s_i s_{i+1}}^A$ and $W_B = \bigcap_{i=m}^k H_{s_i s_{i+1}}^B$.

► **Observation 3.** Let P be a set of k points from $\mathbb{R}\mathcal{P}^2$ in general position and let $\rho \subset \mathbb{R}\mathcal{P}^2$ be an affine plane containing P . A convex set G in $\mathbb{R}\mathcal{P}^2$ is a projective k -gon determined by P if and only if, in ρ , G is either a convex polygon with k vertices (that is, an affine k -gon) or a double chain k -wedge. ◀



■ **Figure 2** A double chain S on 9 points and the corresponding double chain 9-wedge.

► **Observation 4.** Let P be a set of k points from $\mathbb{R}\mathcal{P}^2$ in general position and let $\rho \subset \mathbb{R}\mathcal{P}^2$ be an affine plane containing P . A convex set H in $\mathbb{R}\mathcal{P}^2$ is a projective k -hole in P if and only if, in ρ , H is either a convex polygon with k vertices that is empty in P (that is, an affine k -hole) or a double chain k -wedge that is empty in P . ◀

Convex sets in the real projective plane were considered by many authors [10, 13, 14, 23, 28] and their study goes back more than 100 years to Steinitz [35]. Besides the article of Harborth and Möller [24], which introduced the notion of projective k -gons, we are not aware of any further literature on projective k -gons or projective k -holes. Thus, our goal is to conduct a first extensive study of extremal properties of point sets in $\mathbb{R}\mathcal{P}^2$.

2 Our results

First, we consider an analogue of the Erdős–Szekeres theorem in the real projective plane. For an integer $k \geq 2$, let $ES^p(k)$ be the minimum positive integer N such that every set of at least N points in $\mathbb{R}\mathcal{P}^2$ in general position contains k points in projective convex position. Interestingly, due to Observation 3, $ES^p(k)$ equals the minimum positive integer such that every set of at least $ES^p(k)$ points in \mathbb{R}^2 in general position contains either k points in convex position or a double chain of size k . As already noted in [24], one immediately gets $ES^p(k) \leq ES(k)$. On the other hand, $ES^p(k) \geq ES(\lceil k/2 \rceil)$, since the largest chain of a double chain of size k has at least $\lceil k/2 \rceil$ points. Thus, by (1), we have $2^{\lceil k/2 \rceil - 2} + 1 \leq ES^p(k) \leq 2^{k+O(\sqrt{k \log k})}$ for every $k \geq 2$ and, in particular, the numbers $ES^p(k)$ are finite. As our first result, we prove an almost matching lower bound on $ES^p(k)$.

► **Theorem 5.** There are constants $c, c' > 0$ such that, for every integer $k \geq 2$,

$$2^{k-c \log k} \leq ES^p(k) \leq 2^{k+c'} \sqrt{k \log k}.$$

The precise value of $ES^p(k)$ is known for small values of k . For $k \leq 5$, all sets of k points from $\mathbb{R}\mathcal{P}^2$ determine a projective k -gon by properties (i)–(iii) below Definition 1 and thus $ES^p(k) = k$. Using SAT-solver-based computations, we have also verified the value

$ES^p(6) = 9$, which was determined by Harborth and Möller [24]. This value can also be verified with an exhaustive search, or by using the database of order types of planar point sets [1, 2] or the database of (acyclic) oriented matroids [19, 20]. We also found sets of 17 points from $\mathbb{R}P^2$ with no projective 7-gon, witnessing $ES^p(7) \geq 18$.

Now, we focus on extremal problems about holes in the real projective plane. As our first result, we show that the existence of projective 8-holes is not guaranteed in large point sets in $\mathbb{R}P^2$, proving an analogue of the result by Horton [26].

► **Theorem 6.** *For every $n \in \mathbb{N}$, there exist sets of n points from $\mathbb{R}P^2$ in general position with no projective 8-hole.*

We recall that Theorem 6 implies that there are arbitrarily large finite sets of points from $\mathbb{R}P^2$ in general position with no projective k -holes for any $k \geq 8$. The proof of Theorem 6 uses *Horton sets* defined by Valtr [38] as a generalization of a construction of Horton [26] of an arbitrarily large planar point set in general position (so-called *perfect Horton set*) with no 7-hole; see Section 5 for the definition of Horton sets. Horton sets contain no affine 7-holes in \mathbb{R}^2 and we actually show that, if they are embedded in $\mathbb{R}P^2$, they contain no projective 8-holes. Moreover, we show quadratic bounds on the number of projective k -holes in Horton sets for $k \leq 7$.

► **Theorem 7.** *Let H be a Horton set of size n in $\mathbb{R}^2 \subset \mathbb{R}P^2$. Then H has $\Theta(n^2)$ projective k -holes for every $k \leq 7$. Moreover, if H is the perfect Horton set of size $n = 2^z$, then the number of projective 3-holes in H equals*

$$4.25 \cdot 2^{2z} + 2^z(-3z^2/2 - z/2 - 5.5) - 4z + 2 = 4.25n^2 - 1.5n \log^2 n - \Theta(n \log n).$$

For positive integers $k \geq 3$ and n , let $h_k^p(n)$ be the minimum number of projective k -holes in any set of n points in $\mathbb{R}P^2$ in general position. Theorem 7 gives $h_k^p(n) \leq O(n^2)$ for every $k \leq 7$ and Theorem 6 gives $h_k^p(n) = 0$ for every $k > 7$.

In contrast to the planar case, each sufficiently large Horton set in $\mathbb{R}P^2$ contains a projective 7-hole. We do not have examples of large point sets in $\mathbb{R}P^2$ without projective 7-holes, thus it is natural to ask whether there are projective 7-holes in every sufficiently large point set in $\mathbb{R}P^2$. We believe this to be the case; see Subsection 3 for more open problems.

We also prove that every set of at least 7 points in $\mathbb{R}P^2$ contains a projective 5-hole while there are sets of 6 points in $\mathbb{R}P^2$ with no projective 5-hole. Interestingly, every set of 5 points in $\mathbb{R}P^2$ contains a projective 5-hole. This is in contrast with the situation in the plane, where we have $h_k(n) \leq h_k(n + 1)$ for every k and n , which can be seen by removing a vertex of the convex hull of a set S of $n + 1$ points from \mathbb{R}^2 with $h_k(n + 1)$ affine k -holes.

► **Proposition 8.** *Every set of at least 7 points in general position in $\mathbb{R}P^2$ contains a projective 5-hole. Also, $h_5^p(5) = 1$ and $h_5^p(6) = 0$.*

The proof of Proposition 8 can be found in [7]. The following theorem shows that for some point sets the number of holes is substantially larger in $\mathbb{R}P^2$ than in \mathbb{R}^2 .

► **Theorem 9.** *For every $k \in \{3, \dots, 6\}$ and every positive integer n , there is a set $S_k(n)$ of n points in general position in $\mathbb{R}^2 \subset \mathbb{R}P^2$ such that $S_k(n)$ has $O(n^2)$ affine k -holes in \mathbb{R}^2 and $\Omega(n^{3-\frac{5}{3k}})$ projective k -holes.*

More generally, for every $k \in \{3, \dots, 6\}$, every real number $\alpha \in [0, k - 2]$, and each positive integer n , there is a set $S_k^\alpha(n)$ of n points in general position in $\mathbb{R}^2 \subset \mathbb{R}P^2$ such that $S_k^\alpha(n)$ has $O(n^{2+\alpha})$ affine k -holes in \mathbb{R}^2 and $\Omega(n^{2+\beta})$ projective k -holes, where

$$\beta := \begin{cases} 1 - \frac{5}{3k} + \alpha \cdot \frac{k-1}{k} & \text{if } 0 \leq \alpha \leq \frac{2k-5}{3}, \\ (1 + \alpha) \frac{k-2}{k-1} & \text{if } \frac{2k-5}{3} < \alpha \leq k - 2. \end{cases}$$

10:6 Erdős–Szekeres-Type Problems in the Real Projective Plane

The following result shows a significant difference between the number of holes of all sizes in the plane and in the real projective plane.

► **Theorem 10.** *For any two positive integers n and x with $x \leq 2^{n/2}$, there is a set $S(n, x)$ of n points in general position in $\mathbb{R}^2 \subset \mathbb{RP}^2$ containing at most $O(x + n^2)$ affine holes in \mathbb{R}^2 and at least $\Omega(x^2)$ projective holes.*

In general, we can show that every set P of n points from $\mathbb{R}^2 \subset \mathbb{RP}^2$ contains at least quadratically many projective holes which are not affine holes in \mathbb{R}^2 .

► **Proposition 11.** *Let P be a set of n points in $\mathbb{R}^2 \subset \mathbb{RP}^2$ in general position, and let $h_k^p(P)$ be the number of projective k -holes in P . Then,*

$$h_3^p(P) \geq h_3(P) + \frac{1}{3} \binom{n}{2} \quad \text{and} \quad h_4^p(P) \geq h_4(P) + \frac{1}{2} \left(\binom{n}{2} - 3n + 3 \right),$$

where $h_k(P)$ is the number of affine k -holes in P in the plane \mathbb{R}^2 .

The proof of Proposition 11 can be found in [7]. Together with the best known lower bounds on $h_3(n)$ and $h_4(n)$ by Aichholzer et al. [3], the estimates from Proposition 11 give

$$h_3^p(n) \geq \frac{7}{6}n^2 + \Omega(n \log^{2/3} n) \quad \text{and} \quad h_4^p(n) \geq \frac{3}{2}n^2 + \Omega(n \log^{3/4} n).$$

We also discuss random point sets in the real projective plane and provide the following analogue to results for random point sets in the plane [8, 40]. This gives an alternative proof of the upper bound $h_3^p(n) \leq O(n^2)$. The proof of Theorem 12 can be found in [7].

► **Theorem 12.** *Let K be a compact convex subset in \mathbb{R}^2 of unit area. If P is a set of n points chosen uniformly and independently at random from $K \subset \mathbb{R}^2 \subset \mathbb{RP}^2$, then the expected number of projective 3-holes in P is in $\Theta(n^2)$. Moreover, the expected number of projective holes in P , which are not affine holes in \mathbb{R}^2 , is in $\Theta(n^2)$.*

Last but not least, we discuss the computational complexity of determining the number of k -gons and k -holes in a given point set. Mitchell et al. [30] gave an $O(mn^3)$ time algorithm to compute, for all $k = 3, \dots, m$, the number of k -gons and k -holes in a given set S of n points in the Euclidean plane. Their algorithm also counts k -islands in $O(k^2n^4)$ time. Here, an (affine) k -island in a finite point set S in the plane in general position is the convex hull of a k -tuple I of points from S that does not contain any point from $S \setminus I$. Note that a convex set in \mathbb{R}^2 is a k -hole in S if and only if it is a k -gon and a k -island in S .

Here, we consider the algorithmic aspects of the analogous problems in the real projective plane. By modifying the algorithm by Mitchell et al. [30], we can efficiently compute the number of projective k -gons, k -holes, and k -islands of a finite set in the real projective plane. Here, a projective k -island in a finite set P of points from \mathbb{RP}^2 in general position is a projective convex hull of a k -tuple I of points from P that does not contain any point from $P \setminus I$. Note that, similarly as in the affine case, a convex set in \mathbb{RP}^2 is a projective k -hole in P if and only if it is a projective k -gon and a projective k -island in P .

► **Theorem 13.** *Let P be a set of n points in $\mathbb{R}^2 \subset \mathbb{RP}^2$ in general position. Assuming a RAM model of computation which can perform arithmetic operations on integers in constant time, we can compute the total number of projective k -gons and k -holes in P for $k = 3, \dots, m$ in $O(mn^4)$ time and $O(mn^2)$ space. The number of projective k -islands in P for $k = 3, \dots, m$ can be computed in $O(m^2n^5)$ time and $O(m^2n^3)$ space.*

3 Discussion

The study of extremal questions about finite point sets in $\mathbb{R}\mathcal{P}^2$ suggests a wealth of interesting open problems and topics one can consider. Here, we draw attention to some of them.

By Theorem 6, there are arbitrarily large finite point sets in $\mathbb{R}\mathcal{P}^2$ that avoid k -holes for any $k \geq 8$. On the other hand, the result by Gerken [21] and Nicolas [31] implies that every sufficiently large finite subset of $\mathbb{R}\mathcal{P}^2$ contains a projective k -hole for any $k \leq 6$, as an analogous statement is true already in the affine setting. The existence of projective 7-holes in sufficiently large finite subsets of $\mathbb{R}\mathcal{P}^2$ remains an intriguing open problem and we believe that projective 7-holes can be always found in large points sets in $\mathbb{R}\mathcal{P}^2$.

► **Conjecture 14.** *Every sufficiently large point set in $\mathbb{R}\mathcal{P}^2$ contains a projective 7-hole.*

As we already mentioned, point sets in the plane satisfy $h_k(n) \leq h_k(n+1)$ for all k and n . By Proposition 8, this is no longer true in the real projective plane. However, we do not know any other example violating this inequality except of the single case for 5-holes in $\mathbb{R}\mathcal{P}^2$. Thus, it is natural to ask the following question.

► **Problem 15.** *Is it true that for every integer $k \geq 3$ there is $n_0 = n_0(k)$ such that $h_k^p(n+1) \geq h_k^p(n)$ for every $n \geq n_0$?*

We have shown in Theorem 7 that Horton sets only contain $\Theta(n^2)$ projective k -holes. Since Horton sets only contain $\Theta(n^2)$ affine k -islands [18], which is asymptotically minimal, we wonder whether the same bound applies to projective k -islands.

► **Problem 16.** *For every fixed integer $k \geq 3$, is the minimum number of projective k -islands among all sets of n points from $\mathbb{R}\mathcal{P}^2$ in general position in $\Theta(n^2)$?*

We have shown in Theorem 12 that the expected number of 3-holes in random sets of n points from $\mathbb{R}\mathcal{P}^2$ is in $\Theta(n^2)$. In the plane, we know that the expected number of k -holes and k -islands is in $\Theta(n^2)$ for any fixed k [5, 6]. Can analogous estimates be obtained also in the real projective plane? We note that the lower bound $\Omega(n^2)$ follows from the planar case.

► **Problem 17.** *Let K be a compact convex subset in \mathbb{R}^2 of unit area and let $k \geq 3$. Is the expected number of projective k -holes and k -islands in a set of n points, which is chosen uniformly and independently at random from $K \subset \mathbb{R}^2 \subset \mathbb{R}\mathcal{P}^2$, in $\Theta(n^2)$?*

Besides all these Erdős–Szekeres-type problems related to k -gons, k -holes and k -islands, many other classical problems have natural analogues in the projective plane. In the following, we discuss the problem of *crossing families*. Let P be a finite set of points in the plane. For a positive integer n , let $T(n)$ be the largest number such that any set of n points in general position in the plane determines at least $T(n)$ pairwise crossing segments. The problem of estimating $T(n)$ was introduced in the 1990s by Erdős et al. [4] who proved $T(n) \geq \Omega(\sqrt{n})$. Since then it was widely conjectured that $T(n) \in \Theta(n)$. However, nobody has been able to improve the lower bound from [4] until a recent breakthrough by Pach, Rubin, and Tardos [32] who showed $T(n) \geq n^{1-o(1)}$.

In $\mathbb{R}\mathcal{P}^2$, every pair of points determines a projective line that can be divided into two projective line segments. Given $2n$ points $p_1, \dots, p_k, q_1, \dots, q_k$ from $\mathbb{R}\mathcal{P}^2$, we say that they form *projective crossing family* of size k if, for each i , we can choose a projective line segment s_i between p_i and q_i such that for any pair i, j with $1 \leq i < j \leq k$ the projective line segments s_i and s_j intersect. We can then ask about the maximum size $T^p(n)$ of a projective crossing family in a set P of n points from $\mathbb{R}\mathcal{P}^2$. Note that any set of k pairwise crossing segments of P , which live in a plane $\rho \subset \mathbb{R}\mathcal{P}^2$, gives a projective crossing family of size k in P . Thus, proving a linear lower bound might be simpler for $T^p(n)$ than for $T(n)$.

► **Problem 18.** *Is the maximum size $TP(n)$ of a projective crossing family in a set of n points from \mathbb{RP}^2 in general position in $\Theta(n)$?*

All the notions we discussed (general position, convex position, k -gons, k -holes, k -islands, crossing families, and various others) naturally extend to higher dimensional Euclidean spaces and also to higher dimensional projective spaces. In fact, k -gons and k -holes in higher dimensional Euclidean spaces are currently quite actively studied:

- One central open problem in higher dimensions is to determine the largest value $H(d)$ such that every sufficiently large set in \mathbb{R}^d contains an $H(d)$ -hole. While $H(2) = 6$ is known, the gap between the upper and the lower bound for $H(d)$ remains huge for $d \geq 3$. [11, 12, 34, 39]
- For sets of n points sampled independently and uniformly at random from a unit-volume convex body in \mathbb{R}^d , the expected number of k -holes and k -islands is in $\Theta(n^d)$. [5, 6]
- While the k -gons and k -holes can be counted efficiently in the Euclidean plane, determining the size of the largest gon or hole is NP-hard already in \mathbb{R}^3 . [22]

These analogues in \mathbb{RP}^2 and in high dimensional projective spaces are interesting by themselves, but they might also shed new light on the original problems. We plan to address further such analogues and we hope to also motivate some readers for this line of research.

4 Proof of Theorem 5

Here, we show, for every integer $k \geq 2$, almost matching bounds on the minimum size $ES^p(k)$ that guarantees the existence of a projective k -gon in every set of at least $ES^p(k)$ points from \mathbb{RP}^2 . More precisely, we prove that there are constants $c, c' > 0$ such that

$$2^{k-c \log k} \leq ES^p(k) \leq 2^{k+c'} \sqrt{k \log k}.$$

The upper bound follows from (1), thus it remains to prove the lower bound on $ES^p(k)$. To do so, we construct sets of $2^{k-c \log k}$ points in \mathbb{RP}^2 with no projective k -gon. By Observation 3, it suffices to show that S contains no k points in convex position and no double chain of size k . To obtain such sets, we employ a recursive construction by Erdős and Szekeres [16]. By choosing c sufficiently large, we can assume $k \geq 7$.

Let X and Y be finite sets of points in the Euclidean plane. We say that X *lies deep below* Y and Y *lies high above* X if each point of X lies below every line through two points of Y , and each point of Y lies above every line through two points of X . For $k \geq 2$, we say that a set C of k points in the plane is a k -cup if its points lie on the graph of a convex function and we call C a k -cap if its points lie on the graph of a concave function.

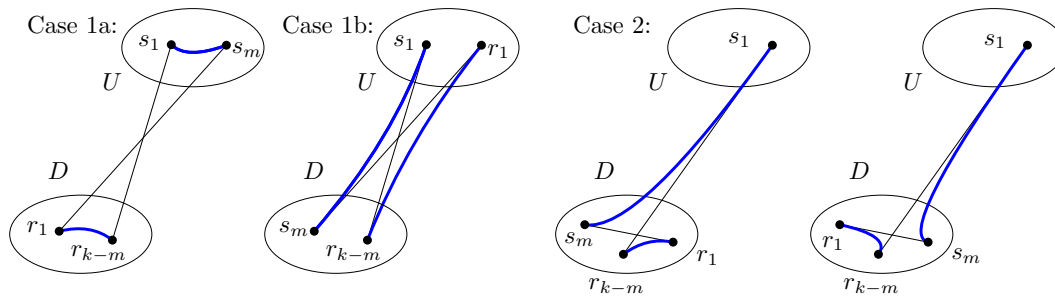
We now construct the set S inductively as follows. For $a \leq 2$ or $u \leq 2$, let $S_{a,u}$ be a set consisting of a single point from \mathbb{R}^2 and note that $S_{a,u}$ then does not contain a 2-cap nor a 2-cup. For integers $a, u \geq 3$, we let $S_{a,u}$ be a set obtained by placing a copy of $S_{a,u-1}$ to the left and deep below a copy of $S_{a-1,u}$. It follows by induction that $|S_{a,u}| = \binom{a+u-4}{a-2} = \binom{a+u-4}{u-2}$ and that $S_{a,u}$ does not contain an a -cap nor a u -cup; see [16]. Finally, we let $S = S_{\lfloor k/2 \rfloor - 1, \lfloor k/2 \rfloor - 1}$. Since $k \geq 7$, we have $\lfloor k/2 \rfloor - 1 \geq 2$ and thus the set S is well-defined.

Note that $|S| = \binom{\lfloor k/2 \rfloor + \lfloor k/2 \rfloor - 4}{\lfloor k/2 \rfloor - 2} \geq 2^{k-c \log k}$ for some constant $c > 0$. The set S does not contain k points in convex position, as such a k -tuple contains either a $(\lfloor k/2 \rfloor - 1)$ -cap or a $(\lfloor k/2 \rfloor - 1)$ -cup. Thus, it remains to show that S does not contain a double chain of size k .

Suppose for contradiction that W is a double chain k -wedge with $A \cup B$ in S with $A = \{s_1, \dots, s_m\}$ and $B = \{r_1, \dots, r_{k-m}\}$ for some m with $1 \leq m \leq k - 1$; using the notation from Subsection 1.2. We let ℓ_1 be the line $\overline{s_1 r_{k-m}}$ and ℓ_2 be the line $\overline{s_m r_1}$. Let

$a \leq \lfloor k/2 \rfloor - 1$ and $u \leq \lfloor k/2 \rfloor - 1$ be two numbers such that W has all vertices in $S_{a,u}$ but it does not have all vertices in $S_{a-1,u}$ nor in $S_{a,u-1}$. Let D and U be the copies of $S_{a-1,u}$ and $S_{a,u-1}$, respectively, forming $S_{a,u}$. We can assume without loss of generality that $|\{s_1, s_m, r_1, r_{k-m}\} \cap D| \geq 2$, as the other case $|\{s_1, s_m, r_1, r_{k-m}\} \cap U| \geq 2$ is treated analogously. We distinguish the following two cases.

Case 1. Assume $|\{s_1, s_m, r_1, r_{k-m}\} \cap D| = 2$. Then two points from $\{s_1, s_m, r_1, r_{k-m}\}$ are in D and the other two points are in U . By symmetry, we can assume $s_1 \in U$. We distinguish the following two subcases, which are shown in Figure 3. Note that, since the line segments s_1r_{k-m} and s_mr_1 cross, the cases $s_1, r_{k-m} \in U$ and $r_1, s_m \in D$ cannot occur.



■ **Figure 3** The cases in the proof of Theorem 5.

Case 1a. Assume $s_1, s_m \in U$ and $r_1, r_{k-m} \in L$. We assume that s_1 is to the left of s_m , otherwise we reverse the order of the elements in A and B which, in particular, exchanges the roles of s_1 and s_m . Since U is high above D , the line $\overline{s_1r_{k-m}}$ is almost vertical and separates s_m from r_1 , where s_1 is to the left of s_m and r_1 is to the left of r_{k-m} . All points of $A \setminus \{s_1\}$ lie to the right of $\overline{s_1r_{k-m}}$ and to the left of $\overline{s_mr_{k-m}}$. Since D is deep below U , no point of D satisfies these two conditions. Hence all points of A lie in U . An analogous argument shows that all points of B lie in D . Since A forms an m -cup in U and B forms a $(k - m)$ -cap in D , we have $m \leq u - 1$ and $k - m \leq a - 1$. Consequently, $k = m + (k - m) \leq a + u - 2 \leq \lfloor k/2 \rfloor + \lfloor k/2 \rfloor - 4 < k$, which is impossible.

Case 1b. Assume $s_1, r_1 \in U$ and $s_m, r_{k-m} \in L$. We assume that s_1 is to the left of r_1 , as otherwise we exchange the roles of A and B which, in particular, exchanges the roles of s_1 and r_1 . Since U is high above D , the line $\overline{s_1r_{k-m}}$ is almost vertical and separates s_m from r_1 and s_m is to the left of r_{k-m} . All points of $A \setminus \{s_1\}$ lie to the left of the almost vertical line $\overline{s_1r_{k-m}}$ and to the right of the almost vertical line $\overline{s_1s_m}$. Hence, $A \cap U = \{s_1\}$ and all points from $A \setminus \{s_1\}$ lie in D . The set $A \setminus \{s_1\}$ forms an $(m - 1)$ -cup in D and thus $m - 1 \leq u - 1$. An analogous argument shows that $B \setminus \{r_1\}$ forms a $(k - m - 1)$ -cap in D and thus $(k - m) - 1 \leq a - 1$. In total, we obtain $k = (m - 1) + (k - m - 1) + 2 \leq (u - 1) + (a - 1) + 2 \leq \lfloor k/2 \rfloor + \lfloor k/2 \rfloor - 2 < k$, which is again impossible.

Case 2. Assume $|\{s_1, s_m, r_1, r_{k-m}\} \cap D| = 3$. We can assume that either s_1 or s_m lies in U , as otherwise we exchange the roles of A and B . Furthermore, we can assume that $s_1 \in U$, as otherwise we reverse the order of the elements in A and B . Since U is high above D , the line $\overline{s_1r_{k-m}}$ is almost vertical and separates r_1 and s_m . Since all vertices of W lie either to the left of the almost vertical line $\overline{s_1s_m}$ and to the right of the almost vertical line $\overline{s_1r_1}$ or to

the right of $\overline{s_1 s_m}$ and to the left of $\overline{s_1 r_1}$, the point s_1 is the only vertex of W in U . Hence, the points $S \setminus \{s_1\}$ lie in D and form an $(m - 1)$ -cup in D . Thus, $m - 1 \leq u$. The points of B all lie in D and form a $(k - m)$ -cap in D . Thus, $k - m \leq a - 1$. Altogether, we have $k = (m - 1) + 1 + (k - m) \leq u + 1 + a - 1 \leq \lfloor k/2 \rfloor + \lfloor k/2 \rfloor - 2 < k$, which is impossible.

Since there is no case left, we have a contradiction with the assumption that W is a double chain k -wedge with vertices in S . This completes the proof of Theorem 5.

5 Sketch of the proofs of Theorem 6 and Theorem 7

Here, we sketch the proof of the fact that there are arbitrarily large finite sets of points from $\mathbb{R}P^2$ in general position with no projective 8-hole and with only quadratically many projective k -holes for every $k \leq 7$. For the full proof see [7].

The construction uses so-called *Horton sets* defined by Valtr [38]. Let H be a set of n points p_1, \dots, p_n from \mathbb{R}^2 , sorted according to increasing x -coordinates. Let H_0 be the set of points p_i with odd i and let H_1 be the set of points p_i with even i . The set H is *Horton* if either $|H| \leq 1$ or if $|H| \geq 2$, H_0 and H_1 are both Horton and H_0 lies deep below or high above H_1 . In the second case, we call H_0 and H_1 the *layers* of H . As in Section 4, we say that H_0 *lies deep below* H_1 and H_1 *lies high above* H_0 if each point of H_0 lies below every line spanned by two points of H_1 , and each point of H_1 lies above every line spanned by two points of H_0 . For a nonempty subset A of H , we define the *base* of A in H as the smallest recursive layer of H containing A .

As in Section 4, we use the terms k -cup and k -cap. A *cap* is a set that is a k -cap for some integer k and, analogously, a *cup* is a set that is a k -cup for some k . A cap C is *open* in a set $S \subseteq \mathbb{R}^2$ if there is no point of S below C , that is, for each pair of points c_1, c_2 from C , no point of S has its coordinate between $x(c_1)$ and $x(c_2)$ and lies below the line $\overline{c_1 c_2}$. Analogously, a cup in S is *open* in S if there is no point of S above it.

5.1 Quadratic upper bounds on the number of k -holes

We show that any Horton set on n points embedded in the real projective plane does not contain 8-holes and that H has at most $O(n^2)$ k -holes for every $k \in \{3, \dots, 7\}$. By Observation 4, it suffices to show that any Horton set H on n points in the plane does not contain 8-holes nor an empty double chain 8-wedge and that, for every $k \in \{3, \dots, 7\}$, H contains only at most $O(n^2)$ k -holes and empty double chain k -wedges. Valtr [38] showed that any Horton set in the plane does not contain 7-holes and that it does not contain any open 4-cap nor an open 4-cup. Bárány and Valtr [9] showed that the number of k -holes in any Horton set of size n is at most $O(n^2)$ for every $k \in \{3, \dots, 6\}$. Thus, it suffices to estimate the number of double chain k -wedges in Horton sets.

Let H be a Horton set with n points in the plane. We first show that the number of open caps in every Horton set H with n points in the plane is at most $O(n)$ and that analogous statement is true for open cups. To prove this claim, it suffices to consider only open 2-caps and 3-caps, as H does not contain open 4-caps.

We proceed by induction on $\log_2 n$ and show that the number $t_2(H)$ of open 2-caps equals $2n - \log_2(n) - 2$ and that the number $t_3(H)$ of open 3-caps in H equals $n - \log_2(n) - 1$ if n is a power of 2. Both expressions hold for $n = 1$ and thus we assume $n \geq 2$. Let p_1, \dots, p_n be the points of H ordered according to increasing coordinates and let $H_0 = L(H)$ and $H_1 = U(H)$ be the sets that partition H such that H_0 is deep below H_1 . Every line segment $p_i p_{i+1}$ forms an open 2-cap in H and there is no other open 2-cap in H with points in H_0

and H_1 , as there is a point of H_1 above any such line segment $p_i p_j$ with $j > i + 1$. Since no two points from H_1 form an open 2-cap in H , we have $t_2(H_0) + n - 1$ open 2-caps in H . By the induction hypothesis, it follows $t_2(H) = 2n - \log_2(n) - 2$.

To determine the number of open 3-caps in H , note that every triple $p_i p_{i+1} p_{i+2}$ with odd i forms an open 3-cap in H . In fact, there is no other open 3-cap in H with a point in H_0 and also in H_1 , as there is a point of H_1 above any such line segment $p_i p_j$ with $j > i + 1$. Since no three points in H_1 form an open 3-cap in H , we obtain $t_3(H_0) + n/2 - 1$ open 3-caps in H . The induction hypothesis then gives $t_3(H) = n - \log_2(n) - 1$.

If n is not a power of two, we consider a Horton set H' of size m instead, where m is as the smallest power of 2 larger than n , and denote its leftmost n points by H'' . Since H'' is also a Horton set of n points and contains the same open caps as H , we obtain $t_2(H) \leq t_2(H') < 4n$ and $t_3(H) \leq t_3(H') < 2n$. Overall, the number of open caps in H is at most $O(n)$. With an analogous argument we obtain the same upper bound on the number of open cups in H .

We now proceed with the proof by induction on n . Clearly, the claims about the double chain k -wedges are true in any Horton set with one or two points, so we assume $n \geq 3$. For some integer $k \geq 3$, let $W \subseteq H$ be a double chain k -wedge that is empty in H . We will show that $k \leq 7$ and estimate the number of such double chain k -wedges for each $k \in \{3, \dots, 7\}$.

If W is contained in H_0 or in H_1 , then $k \leq 7$ by the induction hypothesis. Thus, we assume that W contains a point from H_0 and also from H_1 . An elaborate case analysis shows that H contains no double chain 8-wedge that is empty in H and that has points in H_0 and H_1 ; see [7]. By the induction hypothesis, the sets H_0 and H_1 do not contain any double chain 8-wedge that is empty in H_0 and in H_1 , respectively. Since every double chain 8-wedge that is contained in H_i and is empty in H is also empty in H_i for every $i \in \{0, 1\}$, we see that there is no double chain 8-wedge in H that is empty in H . This completes the proof of Theorem 6.

Let $k \in \{3, \dots, 7\}$. For the quadratic upper bounds, it can be shown that there is a constant c such that H contains at most cn^2 double chain k -wedges that are empty in H and that have points in H_0 and H_1 (again, see [7]). Altogether, the number $w_k(H)$ of empty double chain k -wedges in H satisfies $w_k(H) \leq w_k(H_0) + w_k(H_1) + cn^2$. Solving this linear recurrence with the initial condition $w_k(H') = 0$ for any set H' with $|H'| = 1$ gives $w_k(H) \leq O(n^2)$. This completes the proof of the first part of Theorem 7.

6 Outline of the construction giving Theorems 9 and 10

Here we outline the construction giving Theorems 9 and 10. For the full proof, see [7].

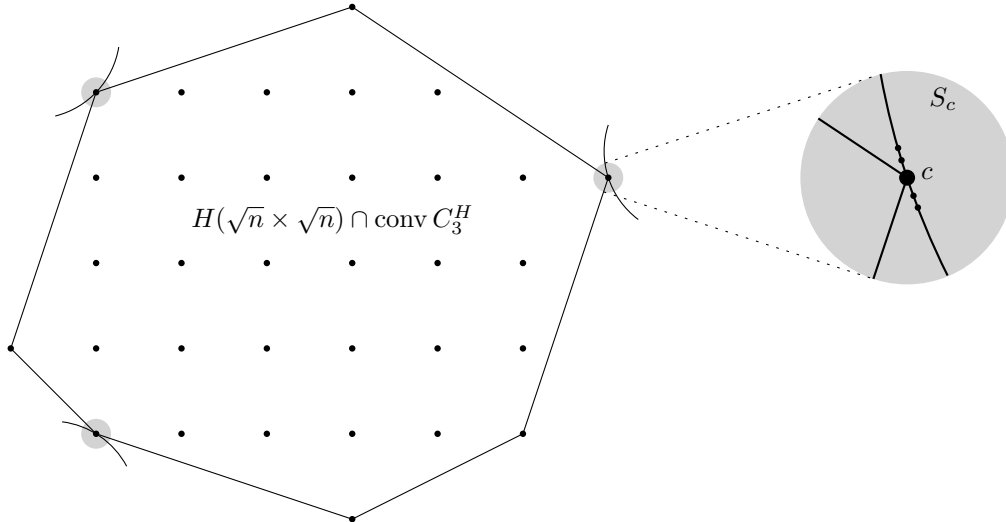
We are given a $k \in \{3, \dots, 6\}$ and a positive integer n . Our construction uses two integer parameters $a, b \geq 2$ satisfying $a \leq n^{1/3}$ and $ab \leq n$. In the proof of Theorem 9, these parameters depend on the value of the parameter α in the theorem. For the proof of Theorem 10, where we are given an integer parameter x , we choose $a := 2$ and $b \approx \log_2(x)$.

Assuming \sqrt{n} is an integer, we start the construction with the $\sqrt{n} \times \sqrt{n}$ integer lattice in the plane, denoted by $L(\sqrt{n} \times \sqrt{n})$, and we fix a subset C_3 of $\Theta(n^{1/3})$ points in convex position in $L(\sqrt{n} \times \sqrt{n})$. We then perturb the lattice to get a so-called *random squared Horton set*, denoted by $H(\sqrt{n} \times \sqrt{n})$, which is a randomized version [9] of the lattice version of so-called Horton sets [38], which generalize the famous construction of Horton [26] of planar point sets in general position with no 7-holes. The random squared Horton set is described in [9, Section 2] and denoted by Λ^* there.

We consider the $|C_3|$ -element subset C_3^H of $H(\sqrt{n} \times \sqrt{n})$ corresponding to C_3 . Since C_3 is in convex position, the set C_3^H is also in convex position. We fix an a -element subset C of C_3^H , where a is the above mentioned parameter. For each $c \in C$, we take a set S_c of b

10:12 Erdős–Szekeres-Type Problems in the Real Projective Plane

points lying in a very small neighborhood of c and on a unit circle touching the polygon $\text{conv } C_3^H$ in the point c . Since the points of S_c are placed very close together on a unit circle, they are almost collinear. We consider the set $H(\sqrt{n} \times \sqrt{n}) \cap \text{conv } C_3^H$, and denote its union with the sets $S_c, c \in C$, by $T = T(a, b)$; see Figure 4. The set T has at most $n + ab \leq 2n$ points, and it is just a little technicality to adjust its size to n at the right place in the proof.



■ **Figure 4** An illustration of the set $T(a, b)$ for $a = 3$ and $b = 5$ (we assume each c lies in S_c).

We now sketch a proof that the set T satisfies Theorems 9 and 10 for properly chosen parameters a and b . The random squared Horton set of size n has $O(n^2)$ affine holes [9, 38]. Likewise, using the condition $ab \leq n$ and two additional facts, it can be argued that the set T has at most $O(n^2)$ affine holes that do not lie completely in some S_c . The two additional facts are that (i) the expected number of affine holes containing a fixed point of C is at most $O(n)$ and (ii) the expected number of affine holes containing a fixed pair of points of C is at most $O(n)$. The number of affine k -holes that lie completely in one of the sets S_c is clearly $a \binom{b}{k} < ab^k$. Thus, the total number of affine k -holes in $T = T(a, b)$ is at most $O(n^2 + ab^k)$.

Due to the construction, any $(k - 1)$ -element subset of any set S_c , together with any point of $T \setminus S_c$, forms a projective k -hole. There are a sets S_c and each of them has size b . Thus, there are at least $a \cdot \binom{b}{k-1} \cdot (|T| - b) = \Theta(ab^{k-1}n)$ projective k -holes in T .

Now, Theorem 9 is obtained from the above construction by setting the parameters a, b carefully with respect to α . Namely, for $\alpha \in [0, \frac{2k-5}{3}]$ we set $a \approx n^{1/3}$ and $b := n^{(5/3+\alpha)/k}$, and for $\alpha \in (\frac{2k-5}{3}, k - 2]$ we set $a \approx n^{1-(1+\alpha)/(k-1)}$ and $b := n^{(1+\alpha)/(k-1)}$. We remark that in the range $\alpha \in [0, \frac{2k-5}{3}]$, the parameter a corresponds to its maximum possible size which is the maximum size of a subset in the lattice $L(\sqrt{n} \times \sqrt{n})$ in convex position, and the parameter b grows with α , since increased α allows bigger affine holes. In the range $\alpha \in (\frac{2k-5}{3}, k - 2]$, the parameter b continues to grow with α but a is decreasing to keep the size ab of S below n .

To obtain Theorem 10 from the above construction, we set $a := 2$ and $b \approx \log_2 x$. Then the number of affine holes contained in one of the two sets S_c is $\approx a2^b = \Theta(x)$ and the number of other affine holes in T is again in $O(n^2)$. Any subset of the $(ab =) 2b$ -element union of the two sets S_c is in convex position or is a double chain, determining a projective hole. Thus, $T = T(2, b)$ has at least $\Theta(2^{2b}) = \Theta(x^2)$ projective holes. Theorem 10 follows.

7 Proof of Theorem 13

Let S be a set of n points in the Euclidean plane in general position. Mitchell et al. [30] use a dynamic programming approach to determine, for every point $p \in S$, the number of k -gons and k -holes for $k = 3, \dots, m$, which have p as the bottom-most point. The algorithm performs in $O(mn^2)$ time and space. They also determine the number of k -islands in S , which have p as the bottom-most point, in $O(m^2n^3)$ time and space. Note that the bottom-most point is unique without loss of generality, as otherwise we perform an affine transformation which does not affect the number of k -gons, k -holes, and k -islands.

Here, we introduce an algorithm that efficiently computes the number of projective k -gons, k -holes, and k -islands of a finite set P of n points from $\mathbb{R}^2 \subset \mathbb{RP}^2$. First, we discuss how to determine the number of projective k -gons in P .

Let G be a projective k -gon with $k \geq 3$ and let p_1, p_2 be two vertices that are consecutive on the boundary of G . If we start at p_1 and trace the boundary of G in the direction of p_2 , we obtain a unique cyclic permutation p_1, \dots, p_k of the vertices of G . By starting at p_2 and tracing in the direction of p_1 , we obtain the reversed cyclic permutation. It is crucial that, independently from the starting point and the direction, only the k pairs $\{p_i, p_{i+1}\}$ for $i = 1, \dots, k$ (indices modulo k) appear as consecutive vertices along the boundary of G .

For every pair of points $\{s, t\} \in P$, the algorithm will count (with multiplicities) the number of projective k -gons in P , which have s and t as consecutive vertices on the boundary. Since each projective k -gon is counted exactly k times, we can then derive the number of projective k -gons in P by a simple division by k .

For a pair $\{s, t\}$ of distinct points from P , we can choose a line $\ell_{s,t}^+$ ($\ell_{s,t}^-$) which is parallel to the line \overline{st} and lies very close and to the left (right) of \overline{st} . By removing $\ell_{s,t}^+$ and $\ell_{s,t}^-$, respectively, from \mathbb{RP}^2 , we obtain two planes $\rho_{s,t}^+ \subset \mathbb{RP}^2$ and $\rho_{s,t}^- \subset \mathbb{RP}^2$. Now, every projective k -gon G of P , which has s and t as consecutive vertices on its boundary, is a convex k -gon either in $\rho_{s,t}^+$ or in $\rho_{s,t}^-$, but not in both. Note that in both planes $\rho_{s,t}^+$ and $\rho_{s,t}^-$, s and t lie on the boundary of the convex hull of P . Moreover, we can assume that s is the bottom-most point in both planes $\rho_{s,t}^+$ and $\rho_{s,t}^-$, as otherwise we apply a suitable rotation.

For each of the $\binom{n}{2}$ pairs $\{s, t\}$ of distinct points from P , we now count the number of convex k -gons in the planes $\rho_{s,t}^+$ and $\rho_{s,t}^-$, which have s and t as consecutive vertices on the boundary. This counting can be done in $O(mn^2)$ time and space by using the algorithm of Mitchell et al. [30] with the slight modification that, in the initial phase, we only count 3-gons of the form $p_1 = s, p_2 = t, p_3$; see equation (3) in [30]. Since each projective k -gon G is now counted precisely k times, once for each pair of consecutive vertices along the boundary of G , this completes the argument for projective k -gons.

Similarly, we count projective k -holes and k -islands. The time and space requirements of the algorithm from [30] for counting projective k -holes, which are incident to the bottom-most point, are the same as for projective k -gons. For counting projective k -islands, which are incident to the bottom-most point, the algorithm from [30] uses $O(m^2n^3)$ time and space.

References

- 1 O. Aichholzer, F. Aurenhammer, and H. Krasser. Data base of order types for small point sets. <http://www.ist.tugraz.at/aichholzer/research/rp/triangulations/order/types/>.
- 2 O. Aichholzer, F. Aurenhammer, and H. Krasser. Enumerating order types for small point sets with applications. *Order*, 19(3):265–281, 2002. doi:10.1023/A:1021231927255.
- 3 O. Aichholzer, M. Balko, T. Hackl, J. Kynčl, I. Parada, M. Scheucher, P. Valtr, and B. Vogtenhuber. A superlinear lower bound on the number of 5-holes. *Journal of Combinatorial Theory, Series A*, 173:Paper No. 105236, 2020. doi:10.1016/j.jcta.2020.105236.

- 4 B. Aronov, P. Erdős, W. Goddard, D. J. Kleitman, M. Klugerman, J. Pach, and L. J. Schulman. Crossing families. *Combinatorica*, 14(2):127–134, 1994.
- 5 M. Balko, M. Scheucher, and P. Valtr. Holes and islands in random point sets. *Random Structures & Algorithms*, 2021. doi:10.1002/rsa.21037.
- 6 M. Balko, M. Scheucher, and P. Valtr. Tight bounds on the expected number of holes in random point sets, 2021. arXiv:2111.12533.
- 7 M. Balko, M. Scheucher, and P. Valtr. Erdős–Szekeres-type problems in the real projective plane, 2022. arXiv:2203.07518.
- 8 I. Bárány and Z. Füredi. Empty simplices in Euclidean space. *Canadian Mathematical Bulletin*, 30(4):436–445, 1987. doi:10.4153/cmb-1987-064-1.
- 9 I. Bárány and P. Valtr. Planar point sets with a small number of empty convex polygons. *Studia Scientiarum Mathematicarum Hungarica*, 41(2):243–266, 2004. doi:10.1556/sscmath.41.2004.2.4.
- 10 J. Bracho and G. Calvillo. Homotopy classification of projective convex sets. *Geometriae Dedicata*, 37:303–306, 1991. doi:10.1007/BF00181406.
- 11 B. Bukh, T. Chao, and R. Holzman. On convex holes in d -dimensional point sets, 2020. arXiv:2007.08972.
- 12 D. Conlon and J. Lim. Fixing a hole. <http://arXiv.org/abs/2108.07087>, 2021. arXiv:2108.07087.
- 13 J. de Groot and H. de Vries. Convex sets in projective space. *Compositio Mathematica*, 13:113–118, 1958. URL: http://www.numdam.org/item/CM_1956-1958__13__113_0/.
- 14 D. Dekker. Convex regions in projective n -space. *The American Mathematical Monthly*, 62(6):430–431, 1955.
- 15 P. Erdős. Some more problems on elementary geometry. *Australian Mathematical Society Gazette*, 5:52–54, 1978. URL: http://www.renyi.hu/~p_erdos/1978-44.pdf.
- 16 P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935. URL: http://www.renyi.hu/~p_erdos/1935-01.pdf.
- 17 P. Erdős and G. Szekeres. On some extremum problems in elementary geometry. *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae, Sectio Mathematica*, 3-4:53–63, 1960.
- 18 R. Fabila-Monroy and C. Huemer. Covering Islands in Plane Point Sets. In *Computational Geometry: XIV Spanish Meeting on Computational Geometry, EGC 2011*, volume 7579 of *Lecture Notes in Computer Science*, pages 220–225. Springer, 2012. doi:10.1007/978-3-642-34191-5_21.
- 19 L. Finschi and K. Fukuda. Generation of oriented matroids—a graph theoretical approach. *Discrete & Computational Geometry*, 27(1):117–136, 2002. doi:10.1007/s00454-001-0056-5.
- 20 Lukas Finschi. Webpage: Homepage of oriented matroids. <http://www.ist.tugraz.at/aichholzer/research/rp/triangulations/orderotypes/>.
- 21 T. Gerken. Empty Convex Hexagons in Planar Point Sets. *Discrete & Computational Geometry*, 39(1):239–272, 2008. doi:10.1007/s00454-007-9018-x.
- 22 P. Giannopoulos, C. Knauer, and D. Werner. On the computational complexity of Erdős–Szekeres and related problems in \mathbb{R}^3 . In *Algorithms – ESA 2013*, pages 541–552. Springer, 2013. doi:10.1007/978-3-642-40450-4_46.
- 23 B. P. Haalmeyer. *Bijdragen tot de theorie der elementaire oppervlakken*. PhD thesis, Amsterdam, 1917.
- 24 H. Harborth and M. Möller. The Esther-Klein-problem in the projective plane. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 15, 1993.
- 25 A. F. Holmsen, H. N. Mojarad, J. Pach, and G. Tardos. Two extensions of the Erdős–Szekeres problem. *Journal of the European Mathematical Society*, 22:3981–3995, 2020. doi:doi/10.4171/JEMS/1000.
- 26 J. D. Horton. Sets with no empty convex 7-gons. *Canadian Mathematical Bulletin*, 26:482–484, 1983. doi:10.4153/CMB-1983-077-8.

- 27 F. Hurtado, M. Noy, and J. Urrutia. Flipping edges in triangulations. *Discrete & Computational Geometry*, 22(3):333–346, 1999. doi:10.1007/PL00009464.
- 28 H. Kneser. Eine Erweiterung des Begriffes “konvexer Körper”. *Mathematische Annalen*, 82(3):287–296, 1921. In German. URL: <https://eudml.org/doc/158850>.
- 29 F. Marić. Fast formal proof of the Erdős–Szekeres conjecture for convex polygons with at most 6 points. *Journal of Automated Reasoning*, 62:301–329, 2019. doi:10.1007/s10817-017-9423-7.
- 30 J. S. B. Mitchell, G. Rote, G. Sundaram, and G. Woeginger. Counting convex polygons in planar point sets. *Information Processing Letters*, 56(1):45–49, 1995. doi:10.1016/0020-0190(95)00130-5.
- 31 C. M. Nicolas. The empty hexagon theorem. *Discrete & Computational Geometry*, 38(2):389–397, 2007. doi:10.1007/s00454-007-1343-6.
- 32 J. Pach, N. Rubin, and G. Tardos. Planar point sets determine many pairwise crossing segments. *Advances in Mathematics*, 386:Paper No. 107779, 2021. doi:10.1016/j.aim.2021.107779.
- 33 M. Scheucher. Two disjoint 5-holes in point sets. *Computational Geometry*, 91:Paper No. 101670, 2020. doi:10.1016/j.comgeo.2020.101670.
- 34 M. Scheucher. A SAT attack on higher dimensional Erdős–Szekeres numbers. In *Extended Abstracts EuroComb 2021*, pages 103–110. Springer, 2021. doi:10.1007/978-3-030-83823-2_17.
- 35 E. Steinitz. Bedingt konvergente Reihen und konvexe Systeme. *Journal für die reine und angewandte Mathematik*, 143:128–176, 1913. In German. URL: <http://eudml.org/doc/149403>.
- 36 A. Suk. On the Erdős–Szekeres convex polygon problem. *Journal of the AMS*, 30:1047–1053, 2017. doi:10.1090/jams/869.
- 37 G. Szekeres and L. Peters. Computer solution to the 17-point Erdős–Szekeres problem. *Australia and New Zealand Industrial and Applied Mathematics*, 48(2):151–164, 2006. doi:10.1017/S144618110000300X.
- 38 P. Valtr. Convex independent sets and 7-holes in restricted planar point sets. *Discrete & Computational Geometry*, 7(2):135–152, 1992. doi:10.1007/bf02187831.
- 39 P. Valtr. Sets in \mathbb{R}^d with no large empty convex subsets. *Discrete Mathematics*, 108(1):115–124, 1992. doi:10.1016/0012-365x(92)90665-3.
- 40 P. Valtr. On the minimum number of empty polygons in planar point sets. *Studia Scientiarum Mathematicarum Hungarica*, pages 155–163, 1995. URL: <https://refubium.fu-berlin.de/handle/fub188/18741>.

True Contraction Decomposition and Almost ETH-Tight Bipartization for Unit-Disk Graphs

Sayan Bandyapadhyay ✉

University of Bergen, Norway

William Lochet ✉

LIRMM, Université de Montpellier, CNRS, France

Daniel Lokshtanov ✉

University of California, Santa Barbara, CA, USA

Saket Saurabh ✉

Institute of Mathematical Sciences, Chennai, India

Jie Xue ✉

New York University Shanghai, China

Abstract

We prove a structural theorem for unit-disk graphs, which (roughly) states that given a set \mathcal{D} of n unit disks inducing a unit-disk graph $G_{\mathcal{D}}$ and a number $p \in [n]$, one can partition \mathcal{D} into p subsets $\mathcal{D}_1, \dots, \mathcal{D}_p$ such that for every $i \in [p]$ and every $\mathcal{D}' \subseteq \mathcal{D}_i$, the graph obtained from $G_{\mathcal{D}}$ by contracting all edges between the vertices in $\mathcal{D}_i \setminus \mathcal{D}'$ admits a tree decomposition in which each bag consists of $O(p + |\mathcal{D}'|)$ cliques. Our theorem can be viewed as an analog for unit-disk graphs of the structural theorems for planar graphs and almost-embeddable graphs proved very recently by Marx et al. [SODA'22] and Bandyapadhyay et al. [SODA'22].

By applying our structural theorem, we give several new combinatorial and algorithmic results for unit-disk graphs. On the combinatorial side, we obtain the first Contraction Decomposition Theorem (CDT) for unit-disk graphs, resolving an open question in the work Panolan et al. [SODA'19]. On the algorithmic side, we obtain a new FPT algorithm for bipartization (also known as odd cycle transversal) on unit-disk graphs, which runs in $2^{O(\sqrt{k} \log k)} \cdot n^{O(1)}$ time, where k denotes the solution size. Our algorithm significantly improves the previous slightly subexponential-time algorithm given by Lokshtanov et al. [SODA'22] (which works more generally for disk graphs) and is almost optimal, as the problem cannot be solved in $2^{o(\sqrt{k})} \cdot n^{O(1)}$ time assuming the ETH.

2012 ACM Subject Classification Theory of computation \rightarrow Design and analysis of algorithms

Keywords and phrases unit-disk graphs, tree decomposition, contraction decomposition, bipartization

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.11

Funding *Daniel Lokshtanov*: Supported by BSF award 2018302 and NSF award CCF-2008838.

Saket Saurabh: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416), and Swarnajayanti Fellowship (No. DST/SJF/MSA01/2017-18).

1 Introduction

For a set \mathcal{D} of unit disks in the plane, the unit-disk graph $G_{\mathcal{D}}$ induced by \mathcal{D} has the unit disks in \mathcal{D} as its vertices, where two vertices are connected by an edge whenever the two corresponding unit disks intersect. As one of the simplest but most important classes of geometric intersection graphs, unit-disk graphs have been extensively studied in various areas (e.g., computational geometry, graph theory, algorithms) and find applications such as modeling the topology of ad-hoc communication networks [27, 49]. The research on unit-disk graphs focused on both combinatorial aspects and algorithmic aspects.



© Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue;

licensed under Creative Commons License CC-BY 4.0

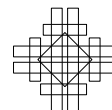
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 11; pp. 11:1–11:16

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



In this paper, we establish a structural theorem for unit-disk graphs, which leads to interesting new results in both combinatorial and algorithmic aspects. Our theorem can be viewed as a unit-disk-graph analog of the very recent theorems proved for planar graphs [39] and more generally for the so-called “almost-embeddable” graphs [5]. Thus, before introducing our theorem, let us first briefly review their results. Specifically, it was shown in [5, 39] that for a planar graph $G = (V, E)$ and a number $p \in [n]$ where $n = |V|$, one can partition V into V_1, \dots, V_p such that for every $i \in [p]$ and $V' \subseteq V_i$, the graph obtained from G by contracting all edges between the vertices in $V_i \setminus V'$ has treewidth $O(p + |V'|)$. Unfortunately, one can easily see that such a statement cannot hold for unit-disk graphs¹. However, if we use the number of *cliques* (instead of vertices) in the bags of the tree decomposition to define its width, this statement is actually true for unit-disk graphs!

Let \mathcal{D} be a set of n unit disks and $p \in [n]$ be any number. Our theorem (roughly) states that one can partition \mathcal{D} into p subsets $\mathcal{D}_1, \dots, \mathcal{D}_p$ such that for every $i \in [p]$ and every $\mathcal{D}' \subseteq \mathcal{D}_i$, the graph obtained from the unit-disk graph $G_{\mathcal{D}}$ by contracting all edges between the vertices in $\mathcal{D}_i \setminus \mathcal{D}'$ admits a tree decomposition in which each bag consists of $O(p + |\mathcal{D}'|)$ cliques. Furthermore, this partition can be computed in polynomial time. The formal statement of our theorem is more technical, and will be presented in Theorem 2 after we introduce some preliminaries in Section 2. Note that the notion of tree decomposition with bags consisting of cliques is not new. In fact, this kind of tree decomposition has been widely applied on unit-disk graphs and other geometric intersection graphs to design efficient algorithms; see for example [12, 21, 43]. In what follows, we discuss the new combinatorial and algorithmic results derived from our main theorem.

Combinatorial application: the first CDT on unit-disk graphs. In graph theory, a *Contraction Decomposition Theorem* (CDT) is a statement of the following form: given a graph $G = (V, E)$ from some graph class, for any $p \in \mathbb{N}$, one can partition E into E_1, \dots, E_p such that contracting the edges in each E_i in G yields a graph of treewidth at most $f(p)$, for some function $f : \mathbb{N} \rightarrow \mathbb{N}$. CDT is classical tool useful in designing efficient approximation and parameterized algorithms in certain classes of graphs. Graph classes for which CDTs are known include planar graphs [31, 32], graphs of bounded genus [15], and H -minor free graphs [14]. However, little was known about CDTs on geometric intersection graphs. Recently, Panolan et al. [44] made the first progress towards proving a CDT for unit-disk graphs. Specifically, they gave a weak version of CDT (which they call a *relaxed CDT*), in which the edge sets E_1, \dots, E_p need not to be disjoint; instead, it is required that each edge $e \in E$ is contained in $O(1)$ sets in E_1, \dots, E_p . It remains open whether unit-disk graphs admit a “true” CDT (where E_1, \dots, E_p is a partition of E). In this paper, by applying our main theorem, we give the first CDT for unit-disk graphs and hence resolve an open question of [44] (and also Hajiaghayi [26]). The function f in our CDT is quadratic, i.e., $f(p) = O(p^2)$, matching the bound in the weak CDT of [44].

Algorithmic application: almost ETH-tight bipartization on unit-disk graphs. Designing efficient algorithms on unit-disk graphs is a central topic in the study of unit-disk graphs. Many classical algorithmic problems have been studied on unit-disk graphs. Polynomial-time solvable problems include shortest paths [7, 8, 47], diameter computing [9, 24], maximum clique [10], etc. Compared to these problems, NP-hard problems received more attentions

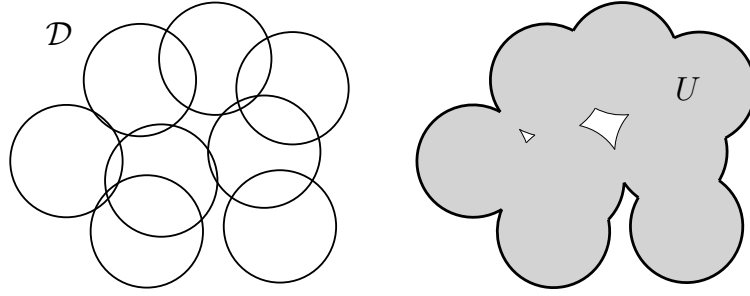
¹ Indeed, the clique K_n is a unit-disk graph, and if we partition the vertices of K_n into p parts for $p \geq 2$, after contracting the smallest part, we get a clique of size at least $n/2$ which has treewidth $\Omega(n)$.

on unit-disk graphs. In particular, studying parametrized algorithms [11] for these hard problems on unit-disk graphs (or other geometric intersection graphs) is one of the most active themes in recent years [2, 3, 20, 21, 22, 23, 43] (also see the survey [44]). A well-known fact about parametrized complexity on planar graphs (or more generally, bounded-genus graphs and H -minor-free graphs) is the so-called “square root phenomenon”: many problems on planar graphs admit algorithms with running time $2^{\tilde{O}(\sqrt{k})}n^{O(1)}$ or $n^{\tilde{O}(\sqrt{k})}$, where k is the parameter (usually the solution size), and also admit (almost) matching lower bounds [6, 13, 16, 18, 19, 33, 34, 40, 42, 46]. Recently, it was shown that such a “square root phenomenon” also appears in many problems on unit disk graphs. Specifically, algorithms with running time $2^{\tilde{O}(\sqrt{k})}n^{O(1)}$ or $n^{\tilde{O}(\sqrt{k})}$ were obtained on unit-disk graphs for VERTEX COVER [12], INDEPENDENT SET [41], FEEDBACK VERTEX SET [4, 20], k -PATH/CYCLE [20, 22], etc. and (almost) matching lower bounds were also known [12]. In this paper, we apply our main theorem to add another classical problem to this family, namely, BIPARTIZATION.

In the BIPARTIZATION problem, one aims to make a graph bipartite by deleting few vertices. Formally, the input of BIPARTIZATION is a graph $G = (V, E)$ and a number k , and the goal is to determine whether there exists $X \subseteq V$ of size at most k such that $G - X$ is *bipartite*. In the literature, BIPARTIZATION is also called ODD CYCLE TRANSVERSAL, as making a graph bipartite is equivalent to hitting all its odd cycles. As one of the most fundamental NP-complete problems in graph theory [48], BIPARTIZATION has been studied extensively over years [1, 17, 25, 28, 29, 30, 35, 45]. The best existing algorithm for BIPARTIZATION on general graphs runs in $2.3146^k n^{O(1)}$ time [36]. On planar graphs, a randomized algorithm with running time $2^{O(\sqrt{k} \log k)} n^{O(1)}$ was known [38, 39], and the same running time was achieved also for bounded-genus graphs and H -minor-free graphs very recently [5]. However, little was known about BIPARTIZATION on geometric intersection graphs. In fact, even achieving *slightly* subexponential-time parameterized algorithm for BIPARTIZATION on unit-disk graphs was a long-standing open problem, prior to the very recent work by Lokshtanov et al. [37]. The authors of [37] gave a randomized algorithm running in $2^{O(k^{\frac{27}{28}} \log k)} n^{O(1)}$ time for BIPARTIZATION on disk graphs (and thus unit-disk graphs), achieving the first $2^{o(k)}$ bound for the problem. This result, however, is still far away from showing the “square root phenomenon” for BIPARTIZATION on unit-disk graphs.

By applying our main theorem, we solve BIPARTIZATION on unit-disk graphs with a randomized algorithm running in $2^{O(\sqrt{k} \log k)} n^{O(1)}$ time, significantly improving the $2^{O(k^{\frac{27}{28}} \log k)}$ bound given by [37]. On the other hand, we establish an almost matching lower bound, showing that the problem cannot be solved in in $2^{o(\sqrt{k})} n^{O(1)}$ time, assuming the Exponential Time Hypothesis (ETH). Our results thus add BIPARTIZATION to the “square root” family of problems on unit-disk graphs. In terms of techniques, our algorithm solves the problem by first constructing the partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of the unit-disk set \mathcal{D} in our main theorem for $p = \sqrt{k}$ and then applying the well-known Baker’s technique on $\mathcal{D}_1, \dots, \mathcal{D}_p$ together with a DP procedure similar to the one in [5] on tree decomposition. Such a scheme based on our theorem can possibly also be applied to solve other problems on unit-disk graphs. To give an example, we extend our algorithm to the problem of GROUP FEEDBACK VERTEX SET with non-identity labels, with the same running time.

Due to limited space, some proofs/details are omitted in this version, and will appear in the full paper.



■ **Figure 1** The boundary and outer boundary of U (the heavier curve is the outer boundary).

2 Preliminaries

The canonical grid. Consider the grid formed by vertical lines $\{x = i : i \in \mathbb{N}\}$ and horizontal lines $\{y = i : i \in \mathbb{N}\}$. We shall use it as the *canonical grid* throughout this paper (in the rest of the paper, we shall refer it as “the grid”). Each cell in the grid is a unit square, and we usually use the notation \square to denote a cell. For a unit disk D , we denote by \square_D the grid cell that contains the center of D . For a set \mathcal{D} of unit disks and a cell \square , we denote by $\mathcal{D} \cap \square$ the subset of unit disks in \mathcal{D} whose centers lie in \square . We say a subset $\mathcal{D}' \subseteq \mathcal{D}$ is *grid-respecting* if for any cell \square such that $\mathcal{D}' \cap \square \neq \emptyset$, we have $\mathcal{D}' \cap \square = \mathcal{D} \cap \square$. A partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of \mathcal{D} is *grid-respecting* if $\mathcal{D}_1, \dots, \mathcal{D}_p$ are all grid-respecting subsets of \mathcal{D} .

Basic graph notions. Let $G = (V, E)$ be a graph. For a subset $V' \subseteq V$, the *subgraph* of G induced by V' is the graph consisting of the vertices in V' and the edges in E with both endpoints in V' . An *induced subgraph* of G is a subgraph of G induced by a subset of V . A vertex $v \in V$ is *neighboring* to a subset $V' \subseteq V$ in G if there exists $v' \in V'$ such that $(v, v') \in E$. A subset $V' \subseteq V$ is *neighboring* to another subset $V'' \subseteq V$ if there exist $v' \in V'$ and $v'' \in V''$ such that $(v', v'') \in E$.

Unit disks and unit-disk graphs. Let \mathcal{D} be a set of unit disks in the plane. For $D \in \mathcal{D}$, we denote by $\text{ctr}(D)$ the *center* of the unit disk D . The union $U = \bigcup_{D \in \mathcal{D}} D$ is a closed region in the plane, whose boundary consists of a set of disjoint closed curves. The *outer boundary* of U is defined as the part of the boundary of U that is incident to the unbounded connected component of $\mathbb{R}^2 \setminus U$; see Figure 1 for an illustration. The *exposed* unit disks in \mathcal{D} refers to the unit disks in \mathcal{D} that intersect the outer boundary of U . In Figure 1, all unit disks in \mathcal{D} are exposed. We denote by $\text{Exp}(\mathcal{D})$ the set of exposed unit disks in \mathcal{D} . The *unit-disk graph* induced by \mathcal{D} , denoted by $G_{\mathcal{D}}$, has the unit disks in \mathcal{D} as its vertices, where two vertices are connected by an edge whenever the two corresponding unit disks intersect. We use $E_{\mathcal{D}}$ to denote the edge set of $G_{\mathcal{D}}$. Note that for a cell \square , the unit disks in $\mathcal{D} \cap \square$ pairwise intersect and hence form a clique in $G_{\mathcal{D}}$, which we call a *cell clique*. We denote by $E_{\mathcal{D}}^* \subseteq E_{\mathcal{D}}$ the set of edges in all cell cliques in $G_{\mathcal{D}}$. For a subset $\mathcal{D}' \subseteq \mathcal{D}$, the unit-disk graph $G_{\mathcal{D}'}$ is canonically isomorphic to the subgraph of $G_{\mathcal{D}}$ induced by \mathcal{D}' . Thus, for convenience, we shall not distinguish between them: we shall also use $G_{\mathcal{D}'}$ to denote the induced subgraph of $G_{\mathcal{D}}$ and use $E_{\mathcal{D}'}$ to denote the set of edges in $G_{\mathcal{D}}$ between the vertices in \mathcal{D}' .

Tree decomposition and treewidth. A *tree decomposition* of a graph $G = (V, E)$ is a pair (T, β) where T is a tree and $\beta : T \rightarrow 2^V$ maps the nodes of T to subsets of V such that (i) $\bigcup_{t \in T} \beta(t) = V$, (ii) for each edge $(u, v) \in E$, there exists $t \in T$ with $u, v \in \beta(t)$, and (iii) for each vertex $v \in V$, the nodes $t \in T$ with $v \in \beta(t)$ form a connected subset in T . Conventionally, we call $\beta(t)$ the *bag* of the node $t \in T$. The *width* of the tree decomposition (T, β) is $\max_{t \in T} |\beta(t)| - 1$. The *treewidth* of a graph G , denoted by $\mathbf{tw}(G)$ is the minimum width of a tree decomposition of G . It is sometimes more convenient to consider *rooted* trees. Thus, throughout this paper, we always view the tree in a tree decomposition as a rooted tree. A tree decomposition (T, β) is *binary* if T is binary.

Edge contraction. From a graph $G = (V, E)$, one can obtain a new graph via a so-called *edge contraction* operation. Specifically, by contracting an edge $e = (u, v) \in E$, we merge u and v into one vertex with edges connecting to both the neighbors of u and the neighbors of v in $V \setminus \{u, v\}$. More generally, we can contract a subset $E_0 \subseteq E$ of edges simply by contracting these edges “one-by-one”. Formally, the resulting graph by contracting E_0 in G , which we denote by G/E_0 , is defined as follows. The vertices of G/E_0 one-to-one corresponds to the connected components of the graph $G_0 = (V, E_0)$, and two vertices have an edge connecting them whenever the corresponding two connected components of G_0 are neighboring in G . Let V_0 denote the vertex set of G/E_0 . Associated to this edge contraction, there is a natural map $\pi : V \rightarrow V_0$ which maps each vertex $v \in V$ to the vertex of G/E_0 corresponding to the connected component of G_0 that contains v . We call π the *quotient map* of the edge contraction. The following fact is a well-known (and can be easily verified).

► **Fact 1.** *Let $G = (V, E)$ be a graph obtained from another graph $G' = (V', E')$ via edge contraction with quotient map $\pi : V' \rightarrow V$. The following statements are true.*

- (i) *If (T, β) is a tree decomposition of G , then (T, β') is a tree decomposition of G' where $\beta'(t) = \pi^{-1}(\beta(t))$ for all nodes $t \in T$.*
- (ii) *If (T', β') is a tree decomposition of G' , then (T', β) is a tree decomposition of G where $\beta(t) = \pi(\beta'(t))$ for all nodes $t \in T'$.*

3 The main theorem

In this section, we present the main theorem of this paper, which establishes a structural property of unit-disk graphs. Formally, the theorem is the following.

► **Theorem 2 (main theorem).** *Given a set \mathcal{D} of n unit disks and an integer $p \in [n]$, one can compute in polynomial time a grid-respecting partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of \mathcal{D} such that for every $i \in [p]$ and every $\mathcal{D}' \subseteq \mathcal{D}_i$, $\mathbf{tw}(G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})) = O(p + |\mathcal{D}'|)$.*

Recall that in Section 1, we gave an informal version of the above theorem, which states that $G_{\mathcal{D}}/E_{\mathcal{D}_i \setminus \mathcal{D}'}$ admits a tree decomposition in which each bag contains $O(p + |\mathcal{D}'|)$ cliques. One may ask how Theorem 2 implies this statement. To see this, observe that $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})$ can be viewed as a graph obtained from $G_{\mathcal{D}}/E_{\mathcal{D}_i \setminus \mathcal{D}'}$ via edge contraction. Thus, if we start from a tree decomposition of $G_{\mathcal{D}}/E_{\mathcal{D}_i \setminus \mathcal{D}'}$ of width $O(p + |\mathcal{D}'|)$ and apply Fact 1 to obtain a tree decomposition of $G_{\mathcal{D}}/E_{\mathcal{D}_i \setminus \mathcal{D}'}$, one can check that each bag of the latter tree decomposition consists of $O(p + |\mathcal{D}'|)$ cliques. We omit the details of this argument as it is not important. The rest of this section is dedicated to proving Theorem 2.

3.1 A layering for the unit disks

The first step of proving Theorem 2 is to compute a *layering* for the unit disks in \mathcal{D} , that is, a decomposition of \mathcal{D} into *layers*. We shall use a function $\ell : \mathcal{D} \rightarrow \mathbb{N}$ to represent the layering: the unit disks which are mapped to i by ℓ form the i -th layer of \mathcal{D} . This layering ℓ *respects* the grid partition of \mathcal{D} in the sense that $\ell^{-1}(\{i\})$ is a grid-respecting subset of \mathcal{D} for all $i \in \mathbb{N}$. Besides, ℓ possesses some nice properties which will be used later to prove Theorem 2. Algorithm 1 presents the procedure for computing ℓ . In words, it iteratively finds the exposed unit disks in \mathcal{D} (line 4) and removes from \mathcal{D} the unit disks whose centers lie in the same cells as the centers of the exposed ones (line 5 and 7), and finally combines the unit disks removed in every 100 iterations into one layer (line 8).

■ **Algorithm 1** LAYERING(\mathcal{D}).

▷ Output a layering $\ell : \mathcal{D} \rightarrow \mathbb{N}$.

```

1:  $q \leftarrow 0$ 
2: while  $\mathcal{D} \neq \emptyset$  do
3:    $q \leftarrow q + 1$ 
4:    $\mathcal{X} \leftarrow \text{Exp}(\mathcal{D})$ 
5:    $\mathcal{X}^+ = \bigcup_{X \in \mathcal{X}} (\mathcal{D} \cap \square_X)$ 
6:    $\text{Tag}_X \leftarrow q$  for all  $X \in \mathcal{X}^+$ 
7:    $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{X}^+$ 
8: return  $\ell : D \mapsto \lceil \text{Tag}_D / 100 \rceil$ 

```

It is clear that the layering ℓ returned by Algorithm 1 respects the cell partition of \mathcal{D} , because in line 6 we always assign the same tag to all unit disks with centers in the cells \square_D . We write $\mathcal{L}_i = \ell^{-1}(\{i\})$ and call it *the i -th layer* of \mathcal{D} . Suppose there are in total m layers. We define $\mathcal{L}_{>i} = \bigcup_{j=i+1}^m \mathcal{L}_j$, $\mathcal{L}_{\geq i} = \bigcup_{j=i}^m \mathcal{L}_j$, $\mathcal{L}_{<i} = \bigcup_{j=1}^{i-1} \mathcal{L}_j$, $\mathcal{L}_{\leq i} = \bigcup_{j=1}^i \mathcal{L}_j$, and $\mathcal{L}_{[i,i']}] = \bigcup_{j=i}^{i'} \mathcal{L}_j$. Next, we establish some nice properties of the layering ℓ .

► **Lemma 3.** *The layering ℓ and the layers $\mathcal{L}_1, \dots, \mathcal{L}_m$ satisfy the following three properties.*

- (i) *For any $D, D' \in \mathcal{D}$ such that $D \cap D' \neq \emptyset$, we have $|\ell(D) - \ell(D')| \leq 1$.*
- (ii) *For a connected component of $G_{\mathcal{L}_{>i}}$ with vertex set $\mathcal{C} \subseteq \mathcal{L}_{>i}$, the unit disks in \mathcal{L}_i neighboring to \mathcal{C} lie in the same connected component of $G_{\mathcal{L}_i}$.*
- (iii) *For any $i, i' \in [m]$ with $i \leq i'$, $\text{tw} \left(G_{\mathcal{L}_{[i,i']}] / E_{\mathcal{L}_{[i,i']}]^*} \right) = O(i' - i + 1)$.*

We remark that the construction of our layering ℓ on unit-disk graphs is analogous to (and also inspired by) the outerplanarity layering on planar graphs (which is obtained by iteratively removing the vertices on the boundary of the outer face of the planar graph). While for the outerplanarity layering the three properties in Lemma 3 follow easily, it requires considerably more work to show them for our layering on unit-disk graphs.

In the rest of this section, we prove Lemma 3. We begin with introducing some notations for ease of exposition. Since \mathcal{D} changes during Algorithm 1, we denote by $\mathcal{D}^{(q)}$ the set \mathcal{D} at the beginning of the q -th iteration of the while-loop (line 2-7). Define $\mathcal{X}^{(q)} = \text{Exp}(\mathcal{D}^{(q)})$ and $U^{(q)}$ as the union of the unit disks in $\mathcal{D}^{(q)}$.

Verifying property (i). Let $D, D' \in \mathcal{D}$ such that $D \cap D' \neq \emptyset$. To show $|\ell(D) - \ell(D')| \leq 1$, it suffices to show $|\text{Tag}_D - \text{Tag}_{D'}| \leq 100$. Let $q = \text{Tag}_D$ and $q' = \text{Tag}_{D'}$. If $q = q'$, we are done. If $q \neq q'$, we may assume $q < q'$ without loss of generality. Since $\text{Tag}_D = q$, $D \in \mathcal{D} \cap \square_X$ for some $X \in \mathcal{X}^{(q)}$. By the definition of $\mathcal{X}^{(q)}$, X intersects the outer boundary of $U^{(q)}$ and

thus there exists a point $x \in X$ that is on the outer boundary of $U^{(q)}$. Let σ be the segment connecting x and $d' = \text{ctr}(D')$. We say a cell \square is *relevant* if there exists a unit disk in $\mathcal{D} \cap \square$ that intersects σ . We observe that there are at least $q' - q + 1$ relevant cells.

► **Observation 4.** *For each $i \in \{q, \dots, q'\}$, there exists a unit disk $D_i \in \mathcal{D}$ with $\text{Tag}_{D_i} = i$ that intersects σ . Thus, the number of relevant cells is at least $q' - q + 1$.*

Note that the length of σ is at most 3 because $D \cap D' \neq \emptyset$ and $D \cap X \neq \emptyset$. As such, there can be no more than 100 relevant cells (actually much fewer), because each relevant cell must contain a point with distance at most 1 from σ . Thus, $q' - q + 1 \leq 100$ and $|\ell(D) - \ell(D')| \leq 1$. Property (i) in Lemma 3 holds.

Verifying property (ii). Consider a connected component of $G_{\mathcal{L}_{>i}}$ with vertex set $\mathcal{C} \subseteq \mathcal{L}_{>i}$. Define $Q = \{q : \lceil q/100 \rceil = i\}$. For a fixed $q \in Q$, the outer boundary of $\mathcal{D}^{(q)}$ consists of some closed curves in the plane, each of which encloses a *region* that is topologically homeomorphic to a disk. These regions are clearly disjoint; we call the union of these regions the *domain* of $\mathcal{D}^{(q)}$. We claim that one of these regions should contain all unit disks in \mathcal{C} . First, observe that the domain of $\mathcal{D}^{(q)}$ contains all unit disks in $\mathcal{D}^{(q)}$, and hence contains all disks in \mathcal{C} since $\mathcal{C} \subseteq \mathcal{L}_{>i} = \mathcal{D}^{(100i+1)} \subseteq \mathcal{D}^{(q)}$. Furthermore, because the regions are disjoint but $G_{\mathcal{C}}$ is connected, all unit disks in \mathcal{C} must lie in the same region. We denote by R_q the region that contains the unit disks in \mathcal{C} . We do this for all $q \in Q$, and thus obtain a set $\{R_q\}_{q \in Q}$ of regions. We observe that these regions are nested.

► **Observation 5.** $R_q \subseteq R_{q'}$ for all $q, q' \in Q$ with $q \geq q'$.

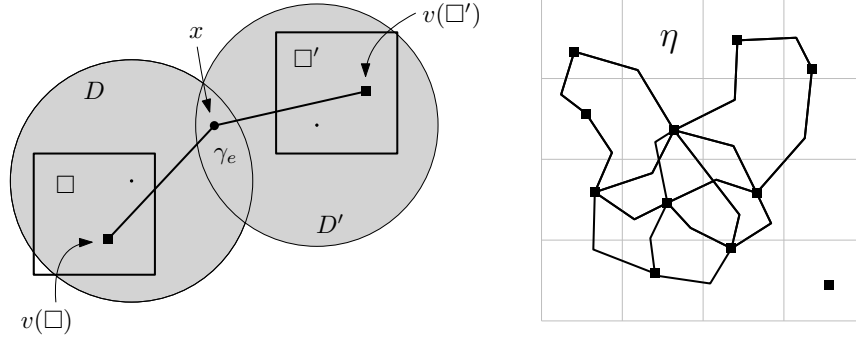
To prove property (ii), consider two unit disks $D, D' \in \mathcal{L}_i$ that are neighboring to \mathcal{C} . Let $q = \text{Tag}_D$ (resp., $q' = \text{Tag}_{D'}$), then the tag of any unit disk in $\mathcal{D} \cap \square_D$ (resp., $\mathcal{D} \cap \square_{D'}$) is q (resp., q'). As $D, D' \in \mathcal{L}_i$, we have $q, q' \in Q$ and we assume $q \geq q'$ without loss of generality. Since D is neighboring to \mathcal{C} and $\text{Tag}_D = q$, D must be contained in R_q and thus all unit disks in $\mathcal{D} \cap \square_D$ are contained in R_q . Furthermore, there exists a unit disk $X \in \mathcal{D} \cap \square_D$ which is exposed in $\mathcal{D}^{(q)}$, i.e., $X \in \mathcal{X}^{(q)}$. Note that X must intersect the boundary of R_q , because X intersects the outer boundary of $U^{(q)}$ and is contained in R_q . Similarly, there exists a unit disk $X' \in \mathcal{D} \cap \square_{D'}$ exposed in $\mathcal{D}^{(q')}$ which intersects the boundary of $R_{q'}$.

► **Observation 6.** $D' \cup X'$ intersects the boundary of R_q .

Now both $D \cup X$ and $D' \cup X'$ are connected and intersect the boundary of R_q . Note that the unit disks in $\mathcal{D}^{(q)}$ that intersect the boundary of R_q form a connected unit-disk graph. Thus, the unit-disk graph induced by these unit disks together with D, X, D', X' is also connected. All these unit disks belong to \mathcal{L}_i , and are hence in the same connected component of $G_{\mathcal{L}_i}$. In particular, D and D' are in the same connected component of $G_{\mathcal{L}_i}$. Property (ii) in Lemma 3 holds.

Verifying property (iii). We notice that, in order to verify property (iii), it suffices to show that $\text{tw}(G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*) = O(j)$ for all $j \in [m]$, because $\mathcal{L}_{[i, i']}$ is nothing but the first $j = i' - i + 1$ layers of the unit-disk set $\mathcal{L}_{\geq i}$. To this end, we first construct a drawing of the graph $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$ on the plane (possibly with edge crossings). The vertices of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$ one-to-one correspond to the cells \square for which $\mathcal{L}_{\leq j} \cap \square \neq \emptyset$, and we denote by $v(\square)$ the vertex corresponding to the cell \square . We draw each vertex $v(\square)$ at an arbitrary point inside the cell \square that lies in the intersection of all unit disks in $\mathcal{D} \cap \square$ (such a point always exists, e.g., the center of \square). For simplicity, we also use $v(\square)$ to denote the point in the plane where

we draw the vertex $v(\square)$. For each edge $e = (v(\square), v(\square'))$ of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$, we draw it as a polyline (or polygonal chain) in the plane connecting $v(\square)$ and $v(\square')$ as follows. Since $v(\square)$ and $v(\square')$ are connected by an edge in $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$, there exist unit disks $D \in \mathcal{L}_{\leq j} \cap \square$ and $D' \in \mathcal{L}_{\leq j} \cap \square'$ such that $D \cap D' \neq \emptyset$. We choose an arbitrary point $x \in D \cap D'$ and let σ be the segment connecting x and $v(\square)$, and σ' be the segment connecting x and $v(\square')$. We then draw the edge e as the two-piece polyline consisting of the segments σ and σ' , and denote this polyline by γ_e . See the left part of Figure 2 for an illustration. Note that γ_e is contained in $D \cup D'$. In this way, we obtain a plane drawing of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$ (possibly with edge crossings), and denote this drawing by η . For convenience, we call the polylines γ_e *edge curves*. Let Γ be the image of η in the plane, which is equal to the union of all edge curves and all $v(\square)$; see the right part of Figure 2. By our construction, Γ is contained in the union of all unit disks in \mathcal{D} . Next, we establish some properties of Γ , which will be used later for bounding $\mathbf{tw}(G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*)$. For two points $a, b \in \mathbb{R}^2$, a *path* from a to b is a continuous map $f : [0, 1] \rightarrow \mathbb{R}^2$ with $f(0) = a$ and $f(1) = b$. We write $\Delta(f, \Gamma) = |\{x \in [0, 1] : f(x) \in \Gamma\}|$; if $\{x \in [0, 1] : f(x) \in \Gamma\}$ is not finite, we simply set $\Delta(f, \Gamma) = \infty$.

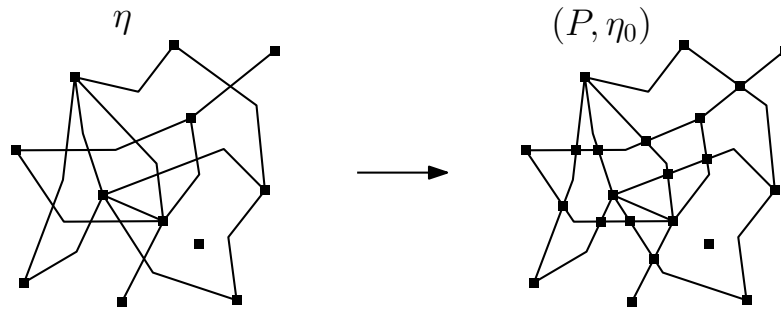


■ **Figure 2** Illustrating the drawing η . The left part is the construction of one edge curve η_e and the right part is an example of how the drawing η finally looks like.

► **Observation 7.** For any two points $a, b \in \mathbb{R}^2$ with distance $O(1)$, there exists a path $f : [0, 1] \rightarrow \mathbb{R}^2$ from a to b such that $\Delta(f, \Gamma) = O(1)$.

► **Observation 8.** For any point $a \in \mathbb{R}^2$, there exists a point b in the unbounded connected component of $\mathbb{R}^2 \setminus \Gamma$ and a path $f : [0, 1] \rightarrow \mathbb{R}^2$ from a to b such that $\Delta(f, \Gamma) = O(j)$.

Proof sketch. We sketch a quick proof of this observation. First, by using Observation 7, one can easily see that for any point $a \in \mathbb{R}^2$, there is a path f from a to a vertex $v(\square)$ of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$ with $\Delta(f, \Gamma) = O(1)$. So it suffices to consider the case where $a = v(\square)$ for some vertex $v(\square)$ of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$. Let q be the tag of the unit disks in $\mathcal{D} \cap \square$. We show the existence of a path f with $\Delta(f, \Gamma) = O(1)$ from $v(\square)$ to some other vertex $v(\square')$ such that the tag of the unit disks in $\mathcal{D} \cap \square'$ is smaller than q . Combining this with a simple induction argument completes the proof of the lemma. There are two cases: there exists such a vertex $v(\square')$ with distance $O(1)$ from $v(\square)$ or there does not exist. In the former case, we directly apply Observation 7 to obtain the path f from $v(\square)$ to $v(\square')$ with $\Delta(f, \Gamma) = O(1)$. In the latter case, we know there is no unit disk in $\mathcal{D} \setminus \mathcal{D}^{(q)}$ that is “close” to $v(\square)$. However, some unit disk in $\mathcal{D} \cap \square$ is exposed in $\mathcal{D}^{(q)}$ but not $\mathcal{D}^{(q-1)}$. That means $v(\square)$ is close to a bounded connected component C of $\mathbb{R}^2 \setminus \bigcup_{D \in \mathcal{D}} D$, which is contained in the unbounded connected component of $\mathbb{R}^2 \setminus U^{(q)}$. In this case, we must have another vertex $v(\square')$ close to C such that



■ **Figure 3** Illustrating the planar graph P obtained by adding vertices to the crossings of η .

the tag of the unit disks in $\mathcal{D} \cap \square'$ is smaller than q . We then construct the path f from $v(\square)$ to $v(\square')$ by first moving from $v(\square)$ into C , then moving inside C to get close to $v(\square')$, and finally moving out from C to $v(\square')$. This summarizes the basic idea of the proof (though the complete proof is more complicated). ◀

The plane drawing η of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$ naturally induces a planar graph P as follows. The vertex set of P consists of the vertices of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$ and the edge-crossing points in the drawing η (called *crossings* for short). Two vertices of P are connected by an edge if they are “adjacent” on some edge curve γ_e . Formally, consider an edge $e = (v(\square), v(\square'))$ of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$. Suppose the crossings on γ_e are c_1, \dots, c_r , ordered from the $v(\square)$ end to the $v(\square')$ end. Then we include in P the edges $(v(\square), c_1), (c_1, c_2), \dots, (c_{r-1}, c_r), (c_r, v(\square'))$. After considering all edges of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$, we complete the construction of P . Note that η naturally induces a planar drawing of P (thus P is planar), which we denote by η_0 . Clearly, the image of η_0 is equal to the image of η , which is Γ . See Figure 3 for an illustration of the construction of P . The following observation gives a relation between the treewidths of $G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*$ and P .

► **Observation 9.** $\text{tw}(G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*) \leq O(\text{tw}(P))$.

Based on the above observation, it now suffices to show that $\text{tw}(P) = O(j)$. To this end, we need to introduce a notion called *vertex-face incidence graph*. We consider the plane-embedded graph (P, η_0) . The *vertex-face incidence graph* P^+ of (P, η_0) is a bipartite graph defined as follows. One part of P^+ consists of the vertices of (P, η_0) , while the other part consists of the faces of (P, η_0) . A vertex v of (P, η_0) and a face F of (P, η_0) are connected by an edge in P^+ if v is incident to F . Let o be the outer face of (P, η_0) , which is a vertex of P^+ . The *depth* of a vertex v in (P, η_0) is defined as the shortest-path distance between o and v in P^+ . It is well-known that $\text{tw}(P)$ is linear in the maximum depth of a vertex in (P, η_0) ; see for example [5]. So we only need to show the depth of each vertex in (P, η_0) is $O(j)$.

Consider a vertex v of (P, η_0) . By Observation 8, there exists a point b in the unbounded connected component of $\mathbb{R}^2 \setminus \Gamma$ and a path $f : [0, 1] \rightarrow \mathbb{R}^2$ from v to b such that $\Delta(f, \Gamma) = O(j)$. Suppose $\{x \in [0, 1] : f(x) \in \Gamma\} = \{x_1, \dots, x_k\}$ where $k = O(j)$ and $x_1 < \dots < x_k$. We have $x_1 = 0$ because $f(0) = v \in \Gamma$. Let $I_i = \{x : x_i < x < x_{i+1}\}$ for $i \in [k - 1]$ and $I_k = \{x : x_k < x \leq 1\}$. Since f is continuous, the image of each I_i under f is connected and disjoint from Γ , and hence lies in one face of (P, η_0) , which we denote by F_i . We say two faces of (P, η_0) are *adjacent* if they are incident to a common vertex of (P, η_0) . Clearly, the shortest-path distance between two adjacent faces of (P, η_0) in P^+ is 2. Note that for each $i \in [k - 1]$, F_i and F_{i+1} are adjacent, as they are both incident to the point $f(x_{i+1}) \in \Gamma$, which is either a vertex of (P, η_0) or on an edge e of (P, η_0) ; in the latter case, F_i and F_{i+1}

are both incident to the two endpoints of e . Therefore, the shortest-path distance between F_1 and F_k in P^+ is at most $2k - 2$, which is $O(j)$. Now F_1 is incident to $f(x_1) = f(0) = v$ and F_k is the outer face o of (P, η_0) since $b \in F_k$. It follows that the shortest-path distance between v and o is $O(j)$, and thus the depth of v is $O(j)$. This implies $\mathbf{tw}(P) = O(j)$ and hence $\mathbf{tw}(G_{\mathcal{L}_{\leq j}}/E_{\mathcal{L}_{\leq j}}^*) = O(j)$ by Observation 9. Property (iii) in Lemma 3 holds.

3.2 Constructing the partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$

Given the layering ℓ of \mathcal{D} presented in the previous section, we are able to construct the partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of \mathcal{D} in Theorem 2. The basic idea is similar to that used in Baker's technique: combining the congruent layers modulo p . Recall that $\mathcal{L}_1, \dots, \mathcal{L}_m$ are the layers of \mathcal{D} . We define $\mathcal{D}_i = \bigcup_{j=0}^{\lfloor (m-i)/p \rfloor} \mathcal{L}_{jp+i}$, i.e., \mathcal{D}_i consists of all layers whose index is congruent to i modulo p . Clearly, $\mathcal{D}_1, \dots, \mathcal{D}_p$ can be computed in polynomial time. As $\{\mathcal{L}_1, \dots, \mathcal{L}_m\}$ is a partition of \mathcal{D} , $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ is also a partition of \mathcal{D} . Also, since each \mathcal{L}_i is a grid-respecting subset of \mathcal{D} , the partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of \mathcal{D} is grid-respecting. To prove Theorem 2, it suffices to show $\mathbf{tw}(G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})) = O(p + |\mathcal{D}'|)$ for any $i \in [p]$ and $\mathcal{D}' \subseteq \mathcal{D}_i$.

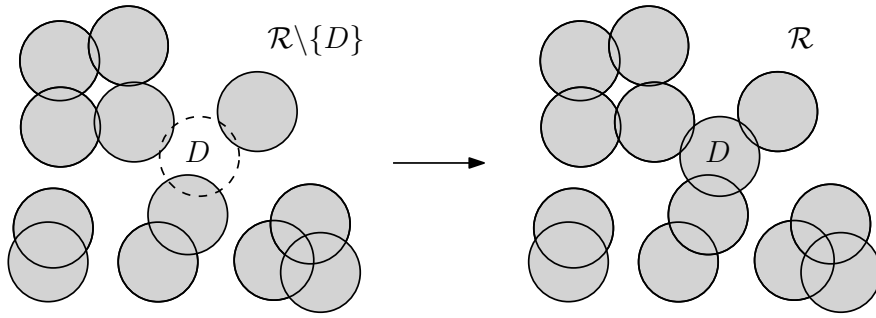
3.3 Bounding the treewidth when $\mathcal{D}' = \emptyset$

We first consider a special case of the treewidth bound in Theorem 2 where $\mathcal{D}' = \emptyset$. In other words, we prove $\mathbf{tw}(G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i})) = O(p)$ for any $i \in [p]$. The argument for this is similar to the one used in [5] for planar graphs. So we only sketch the high-level ideas and the details will appear in the full paper. For simplicity, let us just consider the case $i = p$. Define $r = \lfloor m/p \rfloor + 1$ and $i_j = (j-1) \cdot p$ for $j \in \mathbb{N}$. So we have $\mathcal{D}_p = \bigcup_{j=1}^r \mathcal{L}_{i_j}$. We define a support tree T_{supp} as follows. The depth of T_{supp} is r . The root (i.e., the node at the 0-th level) of T_{supp} is a dummy node. For all $j \in [r]$, the nodes at the j -th level of T_{supp} are one-to-one corresponding to the connected components of $G_{\mathcal{L}_{>i_j}}$. The parent of the nodes at the first level is just the root. Consider a node $t \in T_{\text{supp}}$ at the j -th level for $j \geq 2$. Since $G_{\mathcal{L}_{>i_j}}$ is a subgraph of $G_{\mathcal{L}_{>i_{j-1}}}$, the connected component of $G_{\mathcal{L}_{>i_j}}$ corresponding to t is contained in a unique connected component of $G_{\mathcal{L}_{>i_{j-1}}}$, which corresponds to a node t' at the $(j-1)$ -th level of T_{supp} . We then define the parent of t as t' . For each node $t \in T_{\text{supp}}$, we associate to t a set $\mathcal{A}_t \subseteq \mathcal{D}$ defined as follows. If t is the root, $\mathcal{A}_t = \emptyset$. Suppose t is at the j -th level for $j \in [r]$ and let $\mathcal{C}_t \subseteq \mathcal{L}_{>i_j}$ be the vertex set of the connected component of $G_{\mathcal{L}_{>i_j}}$ corresponding to t . Then we define $\mathcal{A}_t = \{D \in \mathcal{C}_t : i_j < \ell(D) \leq i_{j+1}\}$, i.e., \mathcal{A}_t consists of all unit disks in \mathcal{C}_t which lie in the layers $\mathcal{L}_{i_{j+1}}, \dots, \mathcal{L}_{i_{j+1}}$. We then carefully use the three properties shown in Lemma 3 to argue that $\{\mathcal{A}_t\}_{t \in T_{\text{supp}}}$ is a grid-respecting partition of \mathcal{D} , and $G_{\mathcal{A}_t}$ is adjacent to $G_{\mathcal{A}_{t'}}$ only if t and t' is adjacent in T . Property (iii) implies that each graph $J_t = G_{\mathcal{A}_t}/(E_{\mathcal{A}_t}^* \cup E_{\mathcal{A}_t \cap \mathcal{D}_p})$ has treewidth $O(p)$. Using this fact, we construct an $O(p)$ -width tree decomposition for $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i})$ by “gluing” $O(p)$ -width tree decompositions for the graphs J_t along the edges of T_{supp} . This eventually implies $\mathbf{tw}(G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_p})) = O(p)$.

3.4 Handling the general case

In the previous section, we have proved that the partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ satisfies the condition in Theorem 2 for the special case where $\mathcal{D}' = \emptyset$. In this section, we shall consider the general case and complete the proof for Theorem 2. Let $i \in [p]$. Our goal is to show $\mathbf{tw}(G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})) = O(p + |\mathcal{D}'|)$ for every $\mathcal{D}' \subseteq \mathcal{D}_i$, knowing $\mathbf{tw}(G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i})) = O(p)$.

For convenience, we denote by V the vertex set of $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i})$ and V' the vertex set of $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})$. Since $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i})$ is obtained from $G_{\mathcal{D}}$ via edge contraction, there is a corresponding quotient map $\pi : \mathcal{D} \rightarrow V$. Similarly, there is a quotient map



■ **Figure 4** The three components of $G_{\mathcal{R} \setminus \{D\}}$ hit by D are merged into one connected component in $G_{\mathcal{R}}$, while the others remain the same.

$\pi' : \mathcal{D} \rightarrow V'$ corresponding to the edge contraction for obtaining $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})$. Note that $E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'} \subseteq E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i}$. So there exists a unique map $\rho : V' \rightarrow V$ such that $\pi = \rho \circ \pi'$, and $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i})$ can be viewed as a graph obtained from $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})$ via edge contraction with quotient map ρ .

As $\text{tw}(G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i})) = O(p)$, there exists a tree decomposition (T, β) of $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i})$ of width $O(p)$. We define a map $\beta' : T \rightarrow 2^{V'}$ as $\beta'(t) = \rho^{-1}(\beta(t))$ for all nodes $t \in T$. By Fact 1, (T, β') is a tree decomposition of $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})$. Now it suffices to show that the width of this tree decomposition is $O(p + |\mathcal{D}'|)$. To this end, we establish a basic property of unit-disk graphs. For a graph G , we use the notation $\|G\|$ to denote the number of connected components of G . We have the following lemma.

► **Lemma 10.** *For a set \mathcal{R} of unit disks and $\mathcal{R}' \subseteq \mathcal{R}$, $\|G_{\mathcal{R} \setminus \mathcal{R}'}\| - \|G_{\mathcal{R}}\| = O(|\mathcal{R}'|)$.*

Proof. We show that $\|G_{\mathcal{R} \setminus \{D\}}\| - \|G_{\mathcal{R}}\| = O(1)$ for any unit disk $D \in \mathcal{R}$. Then the lemma can be proved via a simple induction argument. We say D hits a connected component of $G_{\mathcal{R} \setminus \{D\}}$ if D intersects some unit disk in this connected component. Note that all connected components of $G_{\mathcal{R} \setminus \{D\}}$ hit by D are merged into one connected component in $G_{\mathcal{R}}$, and all the other connected components of $G_{\mathcal{R} \setminus \{D\}}$ remain the same in $G_{\mathcal{R}}$. See Figure 4 for an example. Thus, the quantity $\|G_{\mathcal{R} \setminus \{D\}}\| - \|G_{\mathcal{R}}\|$ is equal to the number of connected components of $G_{\mathcal{R} \setminus \{D\}}$ hit by D minus 1. So it suffices to show that D only hits $O(1)$ connected components of $G_{\mathcal{R} \setminus \{D\}}$. Suppose D hits k connected components of $G_{\mathcal{R} \setminus \{D\}}$. Pick a unit disk from each such connected component, and let D_1, \dots, D_k be these unit disks. Note that D_1, \dots, D_k are disjoint as they are from different connected components of $G_{\mathcal{R} \setminus \{D\}}$. On the other hand, D_1, \dots, D_k are all contained in the disk D^+ centered at $\text{ctr}(D)$ of radius 3, as they intersect D . The area of D^+ is 9π , so it can contain at most 9 disjoint unit disks. Thus, $k = O(1)$. ◀

Using the above lemma, we show that $|\rho^{-1}(U)| = O(|U| + |\mathcal{D}'|)$ for any $U \subseteq V$. Since \mathcal{D}_i is a grid-respecting subset of \mathcal{D} , for each $v \in V$, $\pi^{-1}(\{v\})$ is either (the vertex set of) a cell clique of $G_{\mathcal{D}}$ that is disjoint from \mathcal{D}_i or (the vertex set of) a connected component of $G_{\mathcal{D}_i}$; we say v is a *type-1* vertex in the former case and a *type-2* vertex in the latter case. Let U_1 (resp., U_2) be the type-1 (resp., type-2) vertices in U . For each $u \in U_1$, we have $|\rho^{-1}(\{u\})| = |\pi'(\pi^{-1}(\{u\}))| = 1$, as every cell clique of $G_{\mathcal{D}}$ is contracted into one vertex in $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{D}'})$. Thus, $|\rho^{-1}(U_1)| = |U_1|$. To bound $|\rho^{-1}(U_2)|$, we consider $\pi^{-1}(U_2) \subseteq \mathcal{D}$. By definition, $\pi^{-1}(\{u\})$ is a connected component of $G_{\mathcal{D}_i}$ for each $u \in U_2$, and thus $\|G_{\pi^{-1}(U_2)}\| = |U_2|$. Set $\mathcal{I} = \pi^{-1}(U_2) \cap \mathcal{D}'$. By Lemma 10, we have

$$\|G_{\pi^{-1}(U_2) \setminus \mathcal{D}'}\| - \|G_{\pi^{-1}(U_2)}\| = \|G_{\pi^{-1}(U_2) \setminus \mathcal{I}}\| - \|G_{\pi^{-1}(U_2)}\| = O(|\mathcal{I}|),$$

which implies $\|G_{\pi^{-1}(U_2)\setminus\mathcal{D}'}\| = O(|U_2| + |\mathcal{D}'|)$ because $|\mathcal{I}| \leq |\mathcal{D}'|$. Since $\pi^{-1}(U_2)\setminus\mathcal{D}' \subseteq \mathcal{D}_i\setminus\mathcal{D}'$, π' maps the vertices in each connected component of $G_{\pi^{-1}(U_2)\setminus\mathcal{D}'}$ to the same vertex in V' . Therefore, $|\pi'(\pi^{-1}(U_2)\setminus\mathcal{D}')| \leq \|G_{\pi^{-1}(U_2)\setminus\mathcal{D}'}\| = O(|U_2| + |\mathcal{D}'|)$. Now we have the inequality

$$|\pi'(\pi^{-1}(U_2))| \leq |\pi'(\pi^{-1}(U_2)\setminus\mathcal{D}')| + |\pi'(\mathcal{D}')| = O(|U_2| + |\mathcal{D}'|).$$

It follows that $|\rho^{-1}(U_2)| = O(|U_2| + |\mathcal{D}'|)$, and thus $|\rho^{-1}(U)| = O(|U| + |\mathcal{D}'|)$. As a result, for all $t \in T$, $|\beta'(t)| = |\rho^{-1}(\beta(t))| = O(|\beta(t)| + |\mathcal{D}'|) = O(p + |\mathcal{D}'|)$. So (T, β') is a tree decomposition of $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i\setminus\mathcal{D}'})$ of width $O(p + |\mathcal{D}'|)$, completing the proof of Theorem 2.

4 Applications

4.1 Contraction decomposition for unit-disk graphs

In this section, we use Theorem 2 to prove the first Contraction Decomposition Theorem (CDT) for unit-disk graphs, which is shown below.

► **Theorem 11** (Contraction Decomposition Theorem). *Given a set \mathcal{D} of n unit disks and an integer $p \in [n]$, one can compute in polynomial time a partition $\{E_1, \dots, E_p\}$ of $E_{\mathcal{D}}$ such that for every $i \in [p]$, $\mathbf{tw}(G_{\mathcal{D}}/E_i) = O(p^2)$.*

To prove the above theorem, it suffices to compute in polynomial time p disjoint subsets $E_1, \dots, E_p \subseteq E_{\mathcal{D}}$ such that $\mathbf{tw}(G_{\mathcal{D}}/E_i) = O(p^2)$ for every $i \in [p]$ (that is, we do not require $\{E_1, \dots, E_p\}$ to be a partition of $E_{\mathcal{D}}$), as contracting more edges only decreases the treewidth.

We start by applying the algorithm of Theorem 2 on \mathcal{D} to obtain in polynomial time a grid-respecting partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of \mathcal{D} . Consider any $i \in [p]$. Setting $\mathcal{D}' = \emptyset$ in Theorem 2 gives us $\mathbf{tw}(G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i})) = O(p)$. We are going to use this fact later in our analysis. Next, we state a lemma which will be used in our construction of the edge sets E_1, \dots, E_p .

► **Lemma 12.** *The edge set of a clique K of size larger than $4p$ can be partitioned in polynomial time into p parts such that each part contains a spanning tree of K .*

We construct the edge sets E_1, \dots, E_p in the following way. Consider any edge $e = (u, v) \in E_{\mathcal{D}}$. If $u \in \mathcal{D}_i$ and $v \in \mathcal{D}_j$ for $i \neq j$, then we totally ignore e (i.e., do not add it to any of E_1, \dots, E_p). Otherwise, let $u, v \in \mathcal{D}_i$ for some $i \in [p]$. If e is not a part of any cell clique, we add e to the part E_i . If e is a part of a cell clique of size at most $4p$, we also add e to the part E_i . The only remaining edges are those in the cell cliques of size larger than $4p$. Consider any such cell clique K . Using the algorithm in Lemma 12, we partition the edge set of K into exactly p parts H_1, \dots, H_p each of which contains a spanning tree of K , and then add the edges in H_i to E_i for $i \in [p]$. This completes the construction of $E_1, \dots, E_p \subseteq E_{\mathcal{D}}$. It is clear that E_1, \dots, E_p are disjoint. Now it suffices to bound $\mathbf{tw}(G_{\mathcal{D}}/E_i)$ for every $i \in [p]$.

► **Lemma 13.** *For all $i \in [p]$, $\mathbf{tw}(G_{\mathcal{D}}/E_i) = O(p^2)$.*

4.2 Near-optimal bipartization for unit-disk graphs

In this section, we use Theorem 2 to solve BIPARTIZATION on unit-disk graphs. Due to limited space, we only provide a high-level description of our algorithm with details omitted. Let \mathcal{D} be a set of n unit disks and k be the parameter. Recall that we want to find $\mathcal{X} \subseteq \mathcal{D}$ of size at most k such that $G_{\mathcal{D}\setminus\mathcal{X}}$ is bipartite. We refer to such a set \mathcal{X} as an OCT.

An easy but crucial remark is that, for every clique K in $G_{\mathcal{D}}$, an OCT contains all vertices of K except at most two. We start by checking if there is some cell clique in $G_{\mathcal{D}}$ with size at least $k + 3$, in which case it trivially answers NO. From now on, we may assume all cell

cliques have size at most $k + 2$. The first step of our algorithm is to apply the following randomized algorithm to obtain a small candidate set $\text{Cand} \subseteq \mathcal{D}$ for OCT. This can be done via the technique of representative sets, see Lemma 5 in [5] for more details.

► **Lemma 14.** *Given a graph $G = (V, E)$ and a number k , one can compute $\text{Cand} \subseteq V$ of size $k^{O(1)}$ such that G has an OCT of size k iff G has an OCT of size k that is a subset of Cand , using a polynomial-time randomized algorithm with success probability $1 - 1/2^{|V|}$.*

By the above lemma, $|\text{Cand}| = k^{O(1)}$ and it suffices to find an OCT $\mathcal{X} \subseteq \text{Cand}$ of $G_{\mathcal{D}}$ of size at most k . Suppose such an OCT \mathcal{X} exists (but is unknown to us). Next, we apply the algorithm of Theorem 2 with $p = \lfloor \sqrt{k} \rfloor$ to obtain the grid-respecting partition $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ of \mathcal{D} in polynomial time. As $|\mathcal{X}| \leq k$ and $\{\mathcal{D}_1, \dots, \mathcal{D}_p\}$ is a partition of \mathcal{D} , there exists an index $i \in [p]$ such that $|\mathcal{D}_i \cap \mathcal{X}| \leq k/p$. By trying all indices in $[p]$, we can assume that the algorithm knows the index i . Moreover, we know that $\mathcal{D}_i \cap \mathcal{X} \subseteq \mathcal{D}_i \cap \text{Cand}$ as $\mathcal{X} \subseteq \text{Cand}$. Thus, by trying all the subsets of $\mathcal{D}_i \cap \text{Cand}$ of size at most k/p , we can assume that the algorithm knows $\mathcal{S} = \mathcal{D}_i \cap \mathcal{X}$; note that the number of such subsets is $|\text{Cand}|^{O(k/p)} = 2^{O(\sqrt{k} \log k)}$. The above is a variant of Baker's technique, which is also used in [5].

Now it suffices to find an OCT \mathcal{X} of size at most k which contains \mathcal{S} but is disjoint from $\mathcal{D}_i \setminus \mathcal{S}$. By Theorem 2, we have $\text{tw}(G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{S}})) = O(p + |\mathcal{S}|) = O(\sqrt{k})$. Let (T, β^*) be a tree decomposition of $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{S}})$ of width $O(\sqrt{k})$. We can then use Fact 1 to obtain a tree decomposition (T, β) of $G_{\mathcal{D}}$ from (T, β^*) . Then we compute the OCT \mathcal{X} via dynamic programming on (T, β) . The main difficulty here is that although the width of (T, β^*) is $O(\sqrt{k})$, the width of (T, β) is unbounded. Fortunately, we can exploit the $O(\sqrt{k})$ width of (T, β^*) to show that the size of the DP table at each node $t \in T$ as well as the total number of different DP configurations to be considered are both bounded by $2^{O(\sqrt{k} \log k)}$. The main reason is that (essentially) each vertex of $G_{\mathcal{D}}/(E_{\mathcal{D}}^* \cup E_{\mathcal{D}_i \setminus \mathcal{S}})$ corresponds to either a cell clique in $G_{\mathcal{D}}$ or a connected component of $G_{\mathcal{D}_i \setminus \mathcal{S}}$. A cell clique K can have $O(k^2)$ different possible configurations in the solution as by assumption the size of K is $O(k)$ and at most two vertices in K are not in the OCT. A connected component of $G_{\mathcal{D}_i \setminus \mathcal{S}}$ can only have two different configurations as nothing in $\mathcal{D}_i \setminus \mathcal{S}$ is contained in the OCT and a connected graph can have at most two different 2-colorings. As such, we can do DP on (T, β) in $2^{O(\sqrt{k} \log k)} n^{O(1)}$ time despite of its unbounded width. The details of our algorithm will appear in the full paper. Also, the generalization of our algorithm to GROUP FEEDBACK VERTEX SET is deferred to the full version.

► **Theorem 15.** *There exists a randomized algorithm that solves, for given a set \mathcal{D} of n unit disks in the plane and a number k , the BIPARTIZATION problem on $G_{\mathcal{D}}$ in $2^{O(\sqrt{k} \log k)} n^{O(1)}$ time, with success probability at least $1 - 1/2^{|\mathcal{D}|}$.*

We show that the algorithm in the above theorem is near optimal. Specifically, we cannot hope for a $2^{o(\sqrt{k})} n^{O(1)}$ running time, assuming ETH.

► **Theorem 16.** *Assuming the ETH, BIPARTIZATION on unit-disk graphs cannot be solved in $2^{o(\sqrt{k})} n^{O(1)}$ time, where k is the solution size and n is the number of vertices.*

References

- 1 Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $o(\sqrt{\log n})$ approximation algorithms for min uncut, min 2cnf deletion, and directed cut problems. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 573–581, 2005.
- 2 Jochen Alber, Henning Fernau, and Rolf Niedermeier. Graph separators: a parameterized view. *J. Comput. Syst. Sci.*, 67(4):808–832, 2003. doi:10.1016/S0022-0000(03)00072-2.

- 3 Jochen Alber and Jirí Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *J. Algorithms*, 52(2):134–151, 2004. doi:10.1016/j.jalgor.2003.10.001.
- 4 Shinwoo An and Eunjin Oh. Feedback vertex set on geometric intersection graphs. In Hee-Kap Ahn and Kunihiro Sadakane, editors, *32nd International Symposium on Algorithms and Computation, ISAAC 2021, December 6-8, 2021, Fukuoka, Japan*, volume 212 of *LIPICs*, pages 47:1–47:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ISAAC.2021.47.
- 5 Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue. Subexponential parameterized algorithms for cut and cycle hitting problems on h -minor-free graphs. *CoRR*, to appear in *SODA 2022*, abs/2111.14196, 2021. arXiv:2111.14196.
- 6 Thang Nguyen Bui and Andrew Peck. Partitioning planar graphs. *SIAM J. Comput.*, 21(2):203–215, 1992. doi:10.1137/0221016.
- 7 Sergio Cabello and Miha Ježič. Shortest paths in intersection graphs of unit disks. *Computational Geometry*, 48(4):360–367, 2015.
- 8 Timothy M Chan and Dimitrios Skrepetos. All-pairs shortest paths in geometric intersection graphs. In *Workshop on Algorithms and Data Structures*, pages 253–264. Springer, 2017.
- 9 Timothy M Chan and Dimitrios Skrepetos. Approximate shortest paths and distance oracles in weighted unit-disk graphs. *Journal of Computational Geometry (Old Web Site)*, 10(2):3–20, 2019.
- 10 Brent N Clark, Charles J Colbourn, and David S Johnson. Unit disk graphs. *Discrete mathematics*, 86(1-3):165–177, 1990.
- 11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 12 Mark de Berg, Hans L Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C van der Zanden. A framework for h -tight algorithms and lower bounds in geometric intersection graphs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 574–586, 2018.
- 13 Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H -minor-free graphs. *J. ACM*, 52(6):866–893, 2005. doi:10.1145/1101821.1101823.
- 14 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Contraction decomposition in h -minor-free graphs and algorithmic applications. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 441–450, 2011.
- 15 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. *Combinatorica*, 30(5):533–552, 2010.
- 16 Frederic Dorn, Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Beyond bidimensionality: Parameterized subexponential algorithms on directed graphs. *Inf. Comput.*, 233:60–70, 2013. doi:10.1016/j.ic.2013.11.006.
- 17 Samuel Fiorini, Nadia Hardy, Bruce Reed, and Adrian Vetta. Planar graph bipartization in linear time. *Discrete Applied Mathematics*, 156(7):1175–1180, 2008.
- 18 Fedor V. Fomin, Daniel Lokshtanov, Sudeshna Kolay, Fahad Panolan, and Saket Saurabh. Subexponential algorithms for rectilinear steiner tree and arborescence problems. *ACM Trans. Algorithms*, 16(2):21:1–21:37, 2020. doi:10.1145/3381420.
- 19 Fedor V. Fomin, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 515–524. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.62.

- 20 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Decomposition of map graphs with applications. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 60:1–60:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ICALP.2019.60.
- 21 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Finding, hitting and packing cycles in subexponential time on unit disk graphs. *Discret. Comput. Geom.*, 62(4):879–911, 2019. doi:10.1007/s00454-018-00054-x.
- 22 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Eth-tight algorithms for long path and cycle on unit disk graphs. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry, SoCG 2020, June 23-26, 2020, Zürich, Switzerland*, volume 164 of *LIPICs*, pages 44:1–44:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.SoCG.2020.44.
- 23 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Excluded grid minors and efficient polynomial-time approximation schemes. *J. ACM*, 65(2):10:1–10:44, 2018. doi:10.1145/3154833.
- 24 Jie Gao and Li Zhang. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM Journal on Computing*, 35(1):151–169, 2005.
- 25 Michel X Goemans and David P Williamson. Primal-dual approximation algorithms for feedback problems in planar graphs. *Combinatorica*, 18(1):37–59, 1998.
- 26 MohammadTaghi Hajiaghayi. Contraction and minor graph decomposition and their algorithmic applications. *Filmed Talk at Microsoft Research*, 2016.
- 27 William K Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68(12):1497–1514, 1980.
- 28 Falk Hüffner. Algorithm engineering for optimal graph bipartization. *Journal of Graph Algorithms and Applications*, 13(2):77–98, 2009.
- 29 Bart M. P. Jansen, Marcin Pilipczuk, and Erik Jan van Leeuwen. A deterministic polynomial kernel for odd cycle transversal and vertex multiway cut in planar graphs. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPICs*, pages 39:1–39:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.STACS.2019.39.
- 30 Ken-ichi Kawarabayashi and Bruce Reed. An (almost) linear time algorithm for odd cycles transversal. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 365–378. SIAM, 2010.
- 31 Philip N. Klein. A subset spanner for planar graphs, : with application to subset TSP. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 749–756, 2006. doi:10.1145/1132516.1132620.
- 32 Philip N. Klein. A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM J. Comput.*, 37(6):1926–1952, 2008. doi:10.1137/060649562.
- 33 Philip N. Klein and Dániel Marx. Solving planar k -terminal cut in $O(n^{c\sqrt{k}})$ time. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 569–580. Springer, 2012. doi:10.1007/978-3-642-31594-7_48.
- 34 Philip N. Klein and Dániel Marx. A subexponential parameterized algorithm for subset TSP on planar graphs. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1812–1830. SIAM, 2014. doi:10.1137/1.9781611973402.131.

- 35 Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Trans. Algorithms*, 10(4):20:1–20:15, 2014. doi:10.1145/2635810.
- 36 Daniel Lokshtanov, NS Narayanaswamy, Venkatesh Raman, MS Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Transactions on Algorithms (TALG)*, 11(2):1–31, 2014.
- 37 Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. Subexponential parameterized algorithms on disk graphs. *to appear in SODA 2022*, 2021.
- 38 Daniel Lokshtanov, Saket Saurabh, and Magnus Wahlström. Subexponential parameterized odd cycle transversal on planar graphs. In Deepak D’Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPICs*, pages 424–434. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPICs.FSTTCS.2012.424.
- 39 Dániel Marx, Pranabendu Misra, Daniel Neuen, and Prafullkumar Tale. A framework for parameterized subexponential algorithms for generalized cycle hitting problems on planar graphs. *CoRR*, *to appear in SODA 2022*, abs/2110.15098, 2021. arXiv:2110.15098.
- 40 Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. On subexponential parameterized algorithms for steiner tree and directed subset TSP on planar graphs. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 474–484. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00052.
- 41 Dániel Marx and Michal Pilipczuk. Optimal parameterized algorithms for planar facility location problems using voronoi diagrams. In Nikhil Bansal and Irene Finocchi, editors, *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *Lecture Notes in Computer Science*, pages 865–877. Springer, 2015. doi:10.1007/978-3-662-48350-3_72.
- 42 Jesper Nederlof. Detecting and counting small patterns in planar graphs in subexponential parameterized time. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1293–1306. ACM, 2020. doi:10.1145/3357713.3384261.
- 43 Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Contraction decomposition in unit disk graphs and algorithmic applications in parameterized complexity. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1035–1054. SIAM, 2019. doi:10.1137/1.9781611975482.64.
- 44 Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Parameterized computational geometry via decomposition theorems. In Gautam K. Das, Partha Sarathi Mandal, Krishnendu Mukhopadhyaya, and Shin-Ichi Nakano, editors, *WALCOM: Algorithms and Computation - 13th International Conference, WALCOM 2019, Guwahati, India, February 27 - March 2, 2019, Proceedings*, volume 11355 of *Lecture Notes in Computer Science*, pages 15–27. Springer, 2019. doi:10.1007/978-3-030-10564-8_2.
- 45 Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.
- 46 Siamak Tazari. Faster approximation schemes and parameterized algorithms on (odd)-h-minor-free graphs. *Theor. Comput. Sci.*, 417:95–107, 2012. doi:10.1016/j.tcs.2011.09.014.
- 47 Haitao Wang and Jie Xue. Near-optimal algorithms for shortest paths in weighted unit-disk graphs. *Discrete & Computational Geometry*, 64(4):1141–1166, 2020.
- 48 Mihalis Yannakakis. Node-and edge-deletion np-complete problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 253–264, 1978.
- 49 Yu-Shuan Yeh, Joanne C Wilson, and Stuart C Schwartz. Outage probability in mobile telephony with directive antennas and macrodiversity. *IEEE transactions on vehicular technology*, 33(3):123–127, 1984.

Unlabeled Multi-Robot Motion Planning with Tighter Separation Bounds

Bahareh Banyassady ✉ 

Zuse Institute Berlin, Germany

Mark de Berg ✉ 

TU Eindhoven, The Netherlands

Karl Bringmann ✉

Universität des Saarlandes, Saarbrücken, Germany

Max Planck Institute for Informatics, Saarbrücken, Germany

Kevin Buchin ✉ 


TU Dortmund, Germany

Henning Fernau ✉ 

Universität Trier, Germany

Dan Halperin ✉ 

Tel Aviv University, Israel

Irina Kostitsyna ✉ 

TU Eindhoven, The Netherlands

Yoshio Okamoto ✉ 

The University of Electro-Communications, Tokyo, Japan

Stijn Slot ✉

Adyen, Amsterdam, The Netherlands

Abstract

We consider the unlabeled motion-planning problem of m unit-disc robots moving in a simple polygonal workspace of n edges. The goal is to find a motion plan that moves the robots to a given set of m target positions. For the unlabeled variant, it does not matter which robot reaches which target position as long as all target positions are occupied in the end.

If the workspace has narrow passages such that the robots cannot fit through them, then the free configuration space, representing all possible unobstructed positions of the robots, will consist of multiple connected components. Even if in each component of the free space the number of targets matches the number of start positions, the motion-planning problem does not always have a solution when the robots and their targets are positioned very densely. In this paper, we prove tight bounds on how much separation between start and target positions is necessary to always guarantee a solution. Moreover, we describe an algorithm that always finds a solution in time $O(n \log n + mn + m^2)$ if the separation bounds are met. Specifically, we prove that the following separation is sufficient: any two start positions are at least distance 4 apart, any two target positions are at least distance 4 apart, and any pair of a start and a target positions is at least distance 3 apart. We further show that when the free space consists of a single connected component, the separation between start and target positions is not necessary.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases motion planning, computational geometry, simple polygon

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.12

Related Version *Full Version*: <https://arxiv.org/abs/2205.07777>

Funding *Mark de Berg*: Supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.



© Bahareh Banyassady, Mark de Berg, Karl Bringmann, Kevin Buchin, Henning Fernau, Dan Halperin, Irina Kostitsyna, Yoshio Okamoto, and Stijn Slot; licensed under Creative Commons License CC-BY 4.0

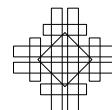
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 12; pp. 12:1–12:16



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Dan Halperin: Supported in part by the Israel Science Foundation (grant no. 1736/19), by NSF/US-Israel-BSF (grant no. 2019754), by the Israel Ministry of Science and Technology (grant no. 103129), by the Blavatnik Computer Science Research Fund, and by the Yandex Machine Learning Initiative for Machine Learning at Tel Aviv University.

Yoshio Okamoto: JSPS KAKENHI Grant Numbers JP20H05795 and JP20K11670.

Acknowledgements This research was initiated at the Lorentz-Center Workshop on Fixed-Parameter Computational Geometry, 2018. We thank Gerhard Woeginger for discussions during the workshop.

1 Introduction

Multi-robot systems are already playing a central role in manufacturing, warehouse logistics, inspection of large structures (e.g., bridges), monitoring of natural resources, and in the future they are expected to expand to other domains such as space exploration, search-and-rescue tasks and more. One of the key ingredients necessary for endowing multi-robot systems with autonomy is the ability to plan collision-free motion paths for their constituent robots towards desired target positions.

In the basic multi-robot motion-planning (MRMP) problem several robots are operating in a common environment. We are given a set of start positions and a set of desired target positions for these robots, and we wish to compute motions that will bring the robots to the targets while avoiding collisions with obstacles and the other robots. We distinguish between two variants of MRMP, *labeled* and *unlabeled*, depending on whether each robot has to reach a specific target. In *labeled* robot motion planning, each robot has a designated target position. In contrast, in the *unlabeled* variant, which we study here, each robot only needs to reach *some* target position; it does not matter which robot reaches which target as long as at the end each target position is occupied by a robot.

MRMP is an extension of the extensively studied *single* robot motion-planning problem (see, e.g., [3, 6, 13]). The multi-robot case is considerably harder [7, 8, 23], since the dimension of the *configuration space* grows with the number of robots in the system. The configuration space of a robot system is a parametric representation of all the possible configurations of the system, which are determined by specifying a real value for each independent parameter (degree of freedom) of the system.

The system we study consists of unit-disc robots moving in the plane; see below for a more formal problem statement. Not only is this a reasonably faithful representation of existing robot systems (e.g., in logistics), but it already encapsulates the essential hardships of MRMP, as MRMP for planar systems with simply-shaped robots are known to be hard [8, 20]. Surprisingly, when we assume some minimum spacing between the start and target positions, the problem for robots moving in a simple polygon always has a solution, and the solution can be found in polynomial time, as shown by Adler et al. [1]. The *separation*, the minimum distance between the start and target positions, thus plays a key role in the difficulty of the problem. However the separation bounds assumed by Adler et al. are not proven to be tight, so the question remains for what separation bounds the problem is always solvable. In this paper we determine the minimal separation needed to ensure that the motion-planning problem has a solution, improving on the bounds obtained by Adler et al. We also describe an algorithm that plans such motions efficiently, relying on the new bounds that we obtain.

Related work. The multi-robot motion planning problem has received much attention over the years. Already in 1983, the problem was described in a paper on the *Piano Mover's problem* by Schwartz and Sharir [15]. Later that year, an algorithm for the case of two or

three disc robots moving in a polygonal environment with n polygon vertices was described, running in $O(n^3)$ and $O(n^{13})$ time respectively [16]. This was later improved by Yap [28] to $O(n^2)$ and $O(n^3)$ for two and three robots respectively, using the *retraction method*. A general approach using *cell decomposition* was later developed in 1991 by Sharir and Sifrony [17] that could deal with a variety of robot pairs in $O(n^2)$ time.

Unfortunately, when the number of robots increases beyond a fixed constant, the problem becomes hard. In 1984, a *labeled* case of the multi-robot motion planning with disc robots and a simple polygonal workspace was shown to be strongly NP-hard [23]. This is a somewhat weaker result than the PSPACE-hardness for many other motion planning problems. For rectangular robots in a rectangular workspace, however, the problem was shown to be PSPACE-hard [8]. This result has later been refined to show that for PSPACE-completeness it is sufficient to have only 1×2 or 2×1 robots in a rectangular workspace [7].

The hardness results for the general problem, as well as the often complex algorithms that solve the problem exactly [6], led to the development of more practical solutions, which often trade completeness of the solution for simplicity and speed, and can successfully cope with motion-planning problems with many degrees of freedom. Most notable among the practical solutions are *sampling-based* (SB) techniques. These include the celebrated Probabilistic Roadmaps (PRM) [9], the Rapidly Exploring Random Trees (RRT) [12], and their numerous variants [3, 5, 13]. The probabilistic roadmaps can be widely applied to explore the high-dimensional configuration space, such as settings with a large number of robots or robots with many degrees of freedom. However, in experiments by Sanchez and Latombe [14] already for 6 robots with a total of 36 degrees of freedom the algorithm requires prohibitively long time to find a solution. Svestka and Overmars [25] suggested an SB algorithm specially tailored to many robots. Their solution still requires exorbitantly large roadmaps and is restricted to a small number of robots. Solovey et al. [21] devised a more economical approach, dRRT (for discrete RRT), which is capable of coping with a larger number of robots, and which was extended to produce asymptotically optimal solutions [18], namely converging to optimal (e.g., shortest overall distance) solution as the number of samples tends to infinity.

Regarding separability bounds, Solomon and Halperin [19] studied the labeled version of the unit-disc problem among polygonal obstacles in the plane, and showed that a solution always exists under a more relaxed separation: each start or target position has an *aura*, namely it resides inside a not-necessarily-concentric disc of radius 2, and the auras of two positions (each being start or target) may overlap, as long as the aura of one robot does not intersect the other robot. They do not however make the distinction between monochromatic and bichromatic separation, and impose the same conditions for all auras.

With respect to unlabeled motion planning, the problem was first considered by Kloder and Hutchinson [10] in 2006. In their paper they provide a sampling-based algorithm which is able to solve the problem. In 2016, Solovey and Halperin [20] have shown that for unit square robots the problem is PSPACE-hard using a reduction from *non-deterministic constraint logic* (NCL) [7]. This PSPACE-hardness result also extends to the labeled variant for unit square robots. Just recently, the unlabeled variant for two classes of disc robots with different radii was also shown to be PSPACE-hard [2], with a similar reduction from NCL. In the reduction the authors use robots of radius $\frac{1}{2}$ and 1. In contrast, the earlier NP-hardness result for disc robots by Spirakis and Yap [23] required discs of many sizes with large differences in radii.

Fortunately, an efficient (polynomial-time) algorithm can still exist when some additional assumptions are made on the problem. Turpin, Michael, and Kumar [26] consider a variant of the unlabeled motion-planning problem where the collection of free positions surrounding every start or target position is star-shaped. This allows them to create an efficient algorithm

for which the path-length is minimized. In the paper by Adler et al. [1], an $O(n \log n + mn + m^2)$ algorithm is given for the unlabeled variant, assuming the workspace is a simple polygon and the start and target positions are *well-separated*, which is defined as minimum distance of four between any start or target position. Their algorithm is based on creating a motion graph on the start and target positions and then treating this as an *unlabeled pebble game*, which can be solved in $O(S^2)$ where S is the number of pebbles [11]. Furthermore, in the paper by Adler et al. [1] the separation bound $4\sqrt{2} - 2$ (≈ 3.646) is shown to be sometimes necessary for the problem to always have a solution. When the workspace contains obstacles, Solovey et al. [22] describe an approximation algorithm which is guaranteed to find a solution when one exists, assuming also that the start and target positions are *well-separated* and a minimum distance of $\sqrt{5}$ between a start or target position and an obstacle.

Finally, we mention that multi-pebble motion on graphs, already brought up above, is part of a large body of work on motion planning in discrete domains, sometimes called multi-agent path finding (MAPF), and often adapted to solving continuous problems; see [24] for a review, and [4, 27, 29, 30] for a sample of recent results.

Contributions. We distinguish between two types of separability bounds: *monochromatic*, denoted by μ , the separation between two start positions or between two target positions, and *bichromatic*, denoted by β , between a start and a target position (see Figure 1).

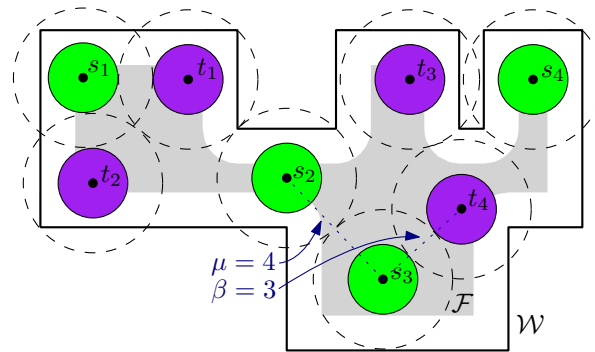
After introducing necessary definitions and notation in Section 2, we begin with a lower bound construction for the monochromatic and bichromatic separation in Section 3. We prove that for $\mu = 4 - \varepsilon$ or for $\beta = 3 - \varepsilon$ (for arbitrarily small positive ε) the solution to the unlabeled multi-robot motion-planning problem in a simple polygon may not always exist.

We devote the remainder of the paper to showing a matching upper bound. We prove that the unlabeled MRMP problem for unit-disc robots in a *simple* polygon is always solvable for monochromatic separation $\mu = 4$ and bichromatic separation $\beta = 3$, assuming that the number of start and target positions match in each free space component. For the case of a single free space component, we show an even stronger result that the problem is always solvable for $\mu = 4$ and $\beta = 0$.

Specifically, in Section 4 we devise an efficient algorithm for MRMP for $\mu = 4$ and $\beta = 2$ in the case of a single free space component, and then extend it to also work for $\mu = 4$ and $\beta = 0$. In Section 5 we extend the algorithm to the case of a free space with multiple components and $\mu = 4$ and $\beta = 3$. Our algorithm runs in $O(n \log n + mn + m^2)$ time, where n is the size of the polygon, and m is the number of robots.

Our results improve upon the results by Adler et al. [1], who describe an algorithm with the same running time that always solves the problem assuming separation of $\mu = \beta = 4$. Similarly to their approach, we restrict the robots to move one at a time on a *motion graph* that has the start and target positions as vertices. Separation of $\mu = \beta = 4$ ensures that the connectivity of the motion graph never changes. However, in our case, the lower bichromatic separation results in a dynamic motion graph: existence of some edges may depend on whether specific nodes are occupied by the robots. Furthermore, the lower bichromatic separation in the case of multiple free space components leads to more intricate dependencies between the components. Nonetheless, we show that there is always an order in which we can process the components, and devise a schedule for the robots to reach their targets.

Due to space restrictions, some proofs are omitted and can be found in the full version of this paper.



■ **Figure 1** Basic definitions. The workspace \mathcal{W} is the rectilinear polygon, the free space \mathcal{F} is the inner gray area. The aura of a start or target position is shown as a dashed circle of radius two (for unit-disc robots). The monochromatic separation $\mu = 4$, the bichromatic separation $\beta = 3$.

2 Definitions and notation

We consider the problem of m indistinguishable unit-disc robots moving in a simple polygonal workspace $\mathcal{W} \subset \mathbb{R}^2$ with n edges. The *obstacle space* \mathcal{O} is the complement of the workspace, that is, $\mathcal{O} = \mathbb{R}^2 \setminus \mathcal{W}$. We refer to points $x \in \mathcal{W}$ as *positions*, and we say that a robot is at position x when its center is positioned at point $x \in \mathcal{W}$.

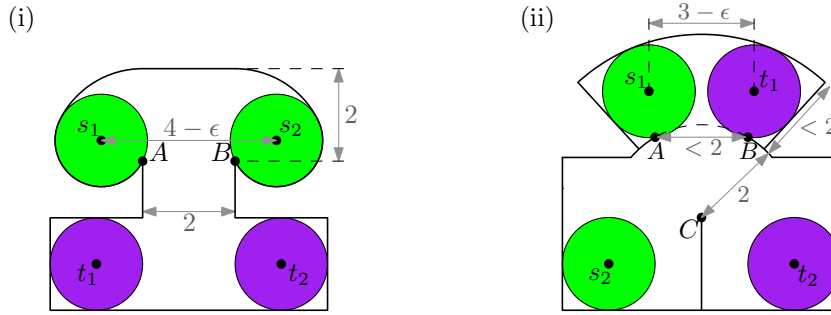
For given $x \in \mathbb{R}^2$ and $r \in \mathbb{R}_+$, we define $\mathcal{D}_r(x)$ to be the open disc of radius r centered at x . The unit-disc robots are defined to be open sets. Thus, a robot collides with the obstacle space \mathcal{O} if and only if its center is at a distance strictly less than 1 from \mathcal{O} . We can now define the *free space* \mathcal{F} to be all positions where a unit-disc robot does not collide with obstacle space, or, more formally, $\mathcal{F} = \{x \in \mathbb{R}^2 \mid \mathcal{D}_1(x) \cap \mathcal{O} = \emptyset\}$. The free space is therefore a closed set. We refer to the connected components of \mathcal{F} as *free space components*.

As the robots are defined to be open sets, two robots collide if the distance between their positions is strictly less than 2. In other words, if a robot occupies a position x then no other robot can be at a position $y \in \mathcal{D}_2(x)$; we call $\mathcal{D}_2(x)$ the *aura* of the robot at position x . In our figures the auras are indicated by dashed circles (see Figure 1).

Unlabeled multi-robot motion-planning problem. Given a set S of m start positions and a set T of m target positions, where $S, T \subset \mathcal{F}$, the goal is to plan a *collision-free* motion for m robots from S to T , such that by the end of the motion every target position in T is occupied by some robot. Since the robots are indistinguishable (i.e., unlabeled), it does not matter which robot ends up at which target position. More formally, we wish to find continuous paths $\pi_i: [0, 1] \rightarrow \mathcal{F}$, for $1 \leq i \leq m$, such that $\pi_i(0) = s_i$ and $\{\pi_i(1) \mid 1 \leq i \leq m\} = T$. Furthermore, we require that, at any moment in time $\tau \in [0, 1]$, for all robots i , no other robot j is in the aura of robot i , $\pi_j(\tau) \notin \mathcal{D}_2(\pi_i(\tau))$. In our figures we indicate start positions by green unit discs centered at points in S , and target positions by purple unit discs centered at points in T .

For a subset $Q \subset \mathcal{F}$ of the free space, we use $S(Q) = S \cap Q$ to denote the set of start positions that reside in Q , and similarly $T(Q) = T \cap Q$ to denote the set of target positions in Q . We define the *charge* $q(Q)$ as the difference between the number of start and target positions in Q , $q(Q) = |S(Q)| - |T(Q)|$. For each free space component F_i , we require that $q(F_i) = 0$, i.e., there needs to be an equal number of start and target positions.

Finally, we state below a few useful properties proven by Adler et al. [1].



■ **Figure 2** (i) An instance for $\mu = 4 - \epsilon$ with one free space component. The robots are blocking each other from entering the corridor. (ii) An instance for $\beta < 3 - \epsilon$. The distance $|AB| < 2$ is too small for a robot to pass through, thus there are two free space components. The robot in the top component is blocking the one in the bottom component.

► **Lemma 1** ([1]). *Each component F_i of the free space is simply connected.*

► **Lemma 2** ([1]). *For any $x \in \mathcal{F}$, let F_i be the connected component of the free space containing x . Then the set $\mathcal{D}_2(x) \cap F_i$ is connected.*

3 Tighter separation bounds

In this section we explore the separation between the start and target positions that is necessary for the problem to always have a solution. We show that, without a certain amount of monochromatic separation (μ) and bichromatic separation (β), there are instances of the problem that cannot be solved, thus certain separation is necessary for the problem to always have a solution. We first prove that a separation of $\mu = 4$ is necessary. This bound is tight and it improves a previous lower bound of $\mu = 4\sqrt{2} - 2 (\approx 3.646)$ [1]. We then show that $\beta = 3$ is also necessary.

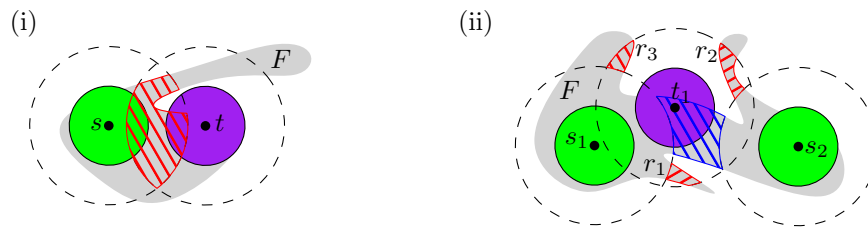
The following lemma is proven using the construction in Figure 2 (i).

► **Lemma 3.** *For $\mu < 4$ a solution does not always exist, even if the free space consists of a single connected component containing two start and two target positions.*

Thus, for a solution to always exist, a monochromatic separation of $\mu = 4$ is necessary. Since the problem for $\mu = \beta = 4$ always has a solution, the monochromatic separation is tight. Hence, we aim to reduce the bichromatic separation β . The proof of the following lemma uses the construction in Figure 2 (ii).

► **Lemma 4.** *For $\beta < 3$ a solution does not always exist, even if there are only two free space components, each containing one start and one target position.*

The lower bound construction for $\beta < 3$ has two free space components with one robot in each. A robot in the top free space component is blocking the motion of a robot in the bottom component, no matter which position it is in. Thus, the lower bound is not applicable if the free space has only one component. Indeed, as we show in the next section, in this case no bichromatic separation is necessary.



■ **Figure 3** (i) When $\beta < 4$, a robot cannot cross the intersection of the auras of s and t (in red) if either s or t is occupied. (ii) $\mathcal{D}_2^-(t_1)$ consists of multiple connected components (remote in red and non-remote in blue). Remote components r_1 and r_2 are blocking areas associated with blocker t_1 .

4 A single free space component

In this section we consider the multi-robot motion-planning problem for the case where the free space consists of a single component F . Initially, for simplicity, we assume $\mu = 4$ and $\beta = 2$. That is, no start/target position can be inside the aura of another start/target position. We later modify the algorithm to handle the case with no bichromatic separation.

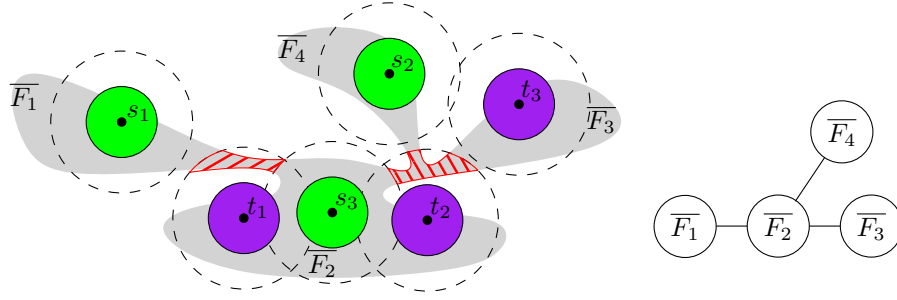
The algorithm by Adler et al. [1] uses the separation assumption $\mu = \beta = 4$, and cannot be applied if $\beta < 4$. Their algorithm greedily moves the robots to the target positions, and may not always be able to find a solution in our case. Indeed, a pair of a start and a target positions whose auras intersect can possibly block the path for robots who need to go through the intersection of these auras (see Figure 3(i)). Therefore, in our algorithm we need to handle such blocking positions.

4.1 Preliminaries

Remote components. Let $A(S) = \bigcup_{s \in S} \mathcal{D}_2(s)$ be the union of all auras of the start positions S . For a target position $t \in T$, let $\mathcal{D}_2^-(t) = (\mathcal{D}_2(t) \cap F) \setminus A(S)$ be the portion of F within the aura of t minus the auras of the start positions in S (see Figure 3(ii)). Note that even though, by Lemma 2, $\mathcal{D}_2(x) \cap F$ is always connected for any $x \in F$, the region $\mathcal{D}_2^-(t)$ may consist of multiple connected components (split by the auras of start positions). One of these components contains t (shown in blue in the figure). A component of $\mathcal{D}_2^-(t)$ that does not contain t is called a *remote component* of t (shown in red in the figure). Let R be the set of remote components for all target positions in T .

Blockers and blocking areas. Consider the example in Figure 3(ii). If t_1 is occupied, its remote components r_1 , r_2 , and r_3 cannot be crossed by a moving robot. Crossing the remote component r_3 can be avoided by moving along its boundary. However remote components r_1 and r_2 pose a problem, as they cut the free space, and thus crossing them cannot be avoided. We call such remote components *blocking areas*.

For a target position $t \in T$, a *blocking area* is a remote component of t that partitions F into multiple components. If t is associated with at least one blocking area, we refer to t as a *blocker*. A blocker might have multiple associated blocking areas (as in Figure 3(ii)). Let $B \subseteq R$ be the set of blocking areas for all target positions in T .



■ **Figure 4** An example with two blockers t_1 and t_2 , and their associated blocking areas shown in red. The corresponding residual components graph H is illustrated on the right side. Note that there is no edge between \overline{F}_4 and \overline{F}_3 , since the blocker t_2 is not located in either of them.

For a blocking area $b \in B$ associated with position t , let the *blocking path* be any path $\pi \subset \mathcal{D}_2(t)$ connecting b to t . By Lemma 2, π exists, and by definition of the blocking area, π crosses the aura of at least one start position. We further show in the following lemma that this path does not intersect any other blocking area.

▶ **Lemma 5.** *For a blocking area $b_x \in B$ and its associated blocker x , there exists some blocking path π such that $\pi \subset \mathcal{D}_2(x)$ and π does not intersect a blocking area b_y of any other blocker y .*

Residual components. Let $\overline{F} = F \setminus \bigcup R$ be the portion of the free space F that does not intersect any remote component in R . By definition, a blocking area partitions F into multiple connected components. Since some remote components are blocking areas, \overline{F} may consist of multiple connected components. We refer to the connected components of \overline{F} as *residual components*. Next, let $F^* = \overline{F} \setminus A(S)$ be the portion of the free space F that does not intersect either the aura of a start position or a remote component of a target position.

▶ **Lemma 6.** *Given m starting and target positions in a polygonal workspace of size n , the subsets \overline{F} and F^* of the free space, and the remote components R , all have complexity $O(m+n)$ and can be computed in $O((m+n)\log(m+n))$ time.*

Residual components graph. We define the *residual components graph* as $H = (V^H, E^H)$ where V^H contains one vertex for each residual component of \overline{F} (see Figure 4). There is an edge between two vertices $v_1, v_2 \in V^H$ if their respective residual components are separated by a single blocking area $b \in B$ and its associated blocker t resides in the respective residual component of either v_1 or v_2 . Although a single blocking area in B can divide \overline{F} into more than two connected components, such a blocking area does not create a cycle in H . This is due to the definition of an edge in H which requires the associated blocker to be in one of the two components. Next lemma follows directly from Lemma 5.

▶ **Lemma 7.** *Any blocking area $b \in B$ shares a boundary with the residual component containing its associated blocker t .*

Next we show that H is a tree by construction.

▶ **Lemma 8.** *The residual components graph H is a tree.*

The general idea of our algorithm is to use the residual components graph H to help us split the problem into smaller subproblems. Using the graph H , we will iteratively choose a leaf residual component \bar{F}_i with a non-positive charge (recall that the charge of a component is the number of start positions minus the number of the target positions), and solve the subproblem restricted to that component using its motion graph, which we define shortly. If afterwards \bar{F}_i will require more robots, they will be moved from a neighboring residual component, ensuring that the blocking area is free for the robots to pass.

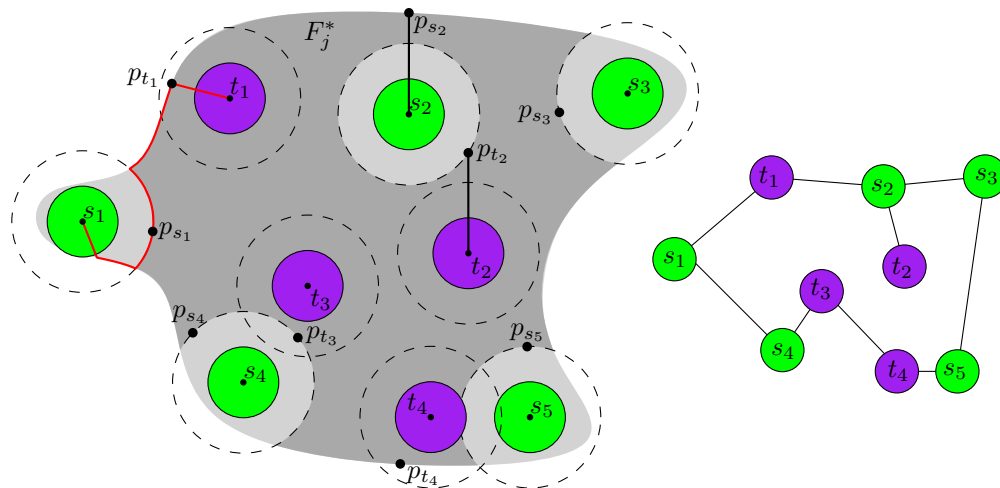
4.2 The motion graph

We now introduce the motion graph, which captures *adjacencies* between the start/target positions. Similarly to [1], the underlying idea of our algorithm is to always have the robots positioned on start or target positions and, using the motion graph, to move one robot at a time between these positions until all target positions are occupied.

Recall that for now we assume that the free space consists of one connected component F . For a free space F with start positions S and target positions T , we define the *motion graph* $G = (V^G, E^G)$, where $V^G = S \cup T$. The edges E^G in G are of two types: *guaranteed* or *blockable*, which we formally define below. Guaranteed edges correspond to so called *guaranteed paths*, where a path in the free space F between $u, v \in S \cup T$ is said to be *guaranteed* if it does not intersect the aura of any position other than u and v .

Unlike guaranteed, blockable edges correspond to paths in F that must cross blocking areas. Our algorithm requires the motion graph G to be connected. However, as $\beta < 4$, without blockable edges the motion graph may be disconnected. Introducing blockable edges ensures that G is connected.

Guaranteed edges. First, we define the guaranteed edges in E^G and show how to construct corresponding guaranteed paths. Recall that we define the set F^* to be the free space minus the auras of the start positions and the remote components, $F^* = F \setminus (\bigcup_{s \in S} \mathcal{D}_2(s) \cup \bigcup R)$.



■ **Figure 5** An illustration of generating the guaranteed edges in a single component F_j^* . Component F_j^* is shown in dark grey; Λ_j is $\langle p_{s_1}, p_{t_1}, p_{s_2}, p_{s_3}, p_{s_5}, p_{t_4}, p_{t_3}, p_{s_4} \rangle$. A path between a pair of adjacent positions is shown in red. The motion graph is shown on the right.

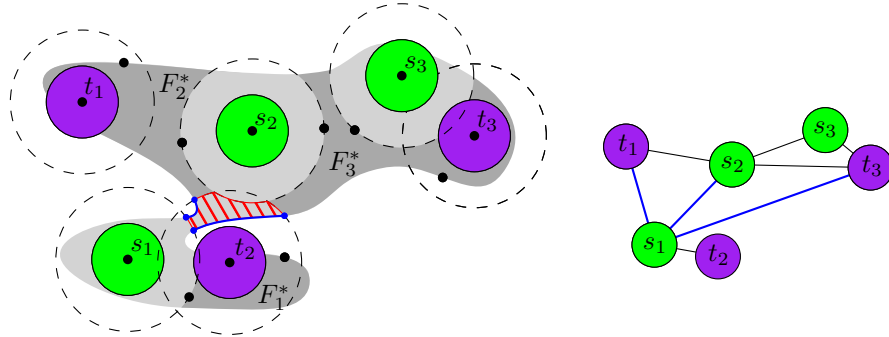
12:10 Unlabeled Multi-Robot Motion Planning with Tighter Separation Bounds

Consider a connected component $F_j^* \subset F^*$. Note that F_j^* may not be simply-connected, as it may contain holes due to subtracted auras of start positions. Abusing the notation, by ∂F_j^* we refer to the outer boundary of F_j^* . For ∂F_j^* , we create an ordered circular list Λ_j of points along ∂F_j^* as follows (see Figure 5).

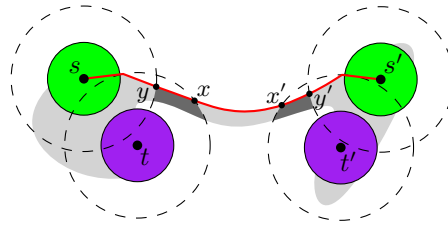
- (i) For each target position $t \in T \cap F_j^*$ whose aura intersects ∂F_j^* , we pick a set of representative points P_t such that P_t contains one point on each connected component of $\partial F_j^* \cap \mathcal{D}_2(t)$. The points P_t are stored in Λ_j based on their ordering along ∂F_j^* .
- (ii) For each position x which is (1) either a target position in F_j^* whose aura does not intersect ∂F_j^* , or (2) a start position corresponding to a hole in F_j^* , we shoot a ray vertically upwards until it hits either ∂F_j^* or the aura of another position y . In the former case the first intersection point p_x is added to Λ_j as a representative point of x . In the latter case a guaranteed edge is added to E^G connecting x and y .
- (iii) Now, consider a start position s whose aura shares a boundary with ∂F_j^* . Note that $\partial F_j^* \cap \mathcal{D}_2(s)$ is connected. If we can pick a representative point p_s on $\partial F_j^* \cap \mathcal{D}_2(s)$ such that there exists an unobstructed path in F connecting s to p_s , then we insert p_s to Λ_j based on its ordering along ∂F_j^* . Otherwise, if for every choice of $p_s \in \partial F_j^* \cap \mathcal{D}_2(s)$ any path connecting s to p_s crosses an aura of some target position t , then we add a guaranteed edge to E^G connecting s and t (for every such target position t). Observe, that by the definition of remote components, if a path connecting s to p_s crosses the aura of t , it must cross it through the non-remote component of t . Thus, there must exist a guaranteed path connecting s and t .

Now that Λ_j is generated, we add a guaranteed edge to the motion graph between any two nodes in G whose representative points are consecutive in Λ_j . If multiple edges between two vertices and self-loops are generated, we remove them in a post-processing step. We repeat this procedure for every connected component $F_j^* \subset F^*$.

Blockable edges. For any blocking area $b \in B$ and its associated blocker t , each section of ∂b is either shared with (1) the boundary of the aura of t , (2) with the boundary of the aura of some start position in S , or (3) with ∂F . See Figure 6 for an illustration. We call a section of ∂b which is shared with ∂F a *free boundary* segment of b . For any free boundary segment of b with endpoints x and y , we assign a set of *incident* positions in $S \cup T$ to x and to y (see below for details). We then add a blockable edge to the motion graph between every pair of incident positions of x and y respectively. Consider an endpoint x of a free boundary segment of b . The set of incident positions of x is defined as follows.



■ **Figure 6** An illustration of a blocking area (in red) with its free boundary (in blue). On the right, the motion graph is shown, with the guaranteed edges (in black) and blockable edges (in blue).



■ **Figure 7** The special case when a blockable edge is added across two blocking areas.

- (i) If x is also an endpoint of a section of ∂b that is shared with $\partial D_2(s)$ for $s \in S$, then s is the only incident position for x .
- (ii) If x is also an endpoint of a section of ∂b that is shared with $\partial D_2(t)$, then x lies on the boundary of a component of F^* . Let that component be F_j^* . Based on the position of x on ∂F_j^* , using Λ_j , we find the predecessor and the successor points of x in Λ_j . By construction of Λ_j , these points are representative of some positions in $S \cup T$. We select those positions as the incident positions for x . The special case that Λ_j is empty, is handled separately and is explained next.

For the special case when Λ_j of F_j^* is empty, if b is the only blocking area incident to ∂F_j^* , then F_j^* does not contain any position in $S \cup T$, and can be ignored. Otherwise, if F_j^* is adjacent to another blocking area b' (of some blocker t'), then from x we follow ∂F_j^* until we reach $\partial b'$ at x' , which must be the endpoint of a free boundary segment of b' (see Figure 7). Let y' be the other endpoint of that free boundary segment. We now select the incident positions of y' as the incident positions of x , i.e., we add a blockable edge between the incident positions of y and those of y' . This results in blockable edges associated with two blocking areas b and b' .

Translating motion graph edges to free space paths. Consider a component $F_j^* \subset F^*$, and let Λ_j be the circular list of representative positions constructed for F_j^* . Let u and v be two positions whose representative points are adjacent in Λ_j . By definition, $(u, v) \in E^G$ is a guaranteed edge, and we claim that there exists a guaranteed path between u and v in F . We construct such a path π_{uv} in the following way.

- (i) First, π_{uv} connects u to its representative point p_u by either following an unobstructed path from u to p_u within u 's aura, or by following the vertical ray used to generate p_u outside of u 's aura.
- (ii) Next, π_{uv} connects p_u to the representative point p_v of v by following ∂F_j^* .
- (iii) Finally, π_{uv} connects p_v to v similarly to (i).

Now consider the case when a guaranteed edge (u, v) is constructed without adding the representative points to Λ_j . If (u, v) is constructed according to the case (ii) of the definition of the guaranteed edges, and without loss of generality the vertical ray emanates from u , then the guaranteed path π_{uv} consists of the vertical segment up_u and an unobstructed path connecting the representative point p_u to the node v within the aura of v . If (u, v) is constructed in case (iii), and without loss of generality $u \in S$ and $v \in T$, then the guaranteed path π_{uv} consists of a path from u until the first intersection with the aura of v and an unobstructed path to v within its aura.

The paths for blockable edges are constructed in the following way. Each endpoint of a free boundary segment is incident to at most two representative points in some list Λ_i . Thus, each free boundary segment (x, y) of every blocking area b contributes up to four edges to the motion graph. Consider a blockable edge (u_x, v_y) between an incident position u_x of x and an incident position v_y of y . The corresponding path consists of three parts.

- (i) From u_x to x . This part is generated similarly to the part (i) for guaranteed edges.
- (ii) From x to y . This part follows the free boundary of b between x and y .
- (iii) From y to v_y . This part is generated similarly to the part (iii) for guaranteed edges.

The following proposition, lists five properties of the motion graph, which will be used to derive the correctness and the complexity of the algorithm.

► **Proposition 9.** *The following properties of a motion graph G hold.*

1. *There exists a guaranteed path in F for each guaranteed edge in G .*
2. *There exists a path in G consisting solely of guaranteed edges between any two positions inside the same residual component $\bar{F}_j \in \bar{F}$.*
3. *G is connected.*
4. *The number of edges $|E^G|$ in G is bounded by $O(m)$.*
5. *Between any two vertices of G , we can find a path in $O(m)$ time, and the corresponding path in the free space has complexity of $O(m + n)$.*
6. *The motion graph G can be created in $O(mn + m^2)$ time.*

4.3 The algorithm

We are now ready to describe our algorithm. We use the residual components graph H , which is a tree, in order to split the problem into smaller subproblems, and recursively solve them. Using H we select a particular residual component of the free space, and solve the subproblem restricted to it using the motion graph. Proposition 9 will help us ensure that such reconfiguration is always possible. One key point is to select a vertex of H , such that, after solving the subproblem in the corresponding residual component, no robots need to move across the incident blocking areas. That is, we need to choose the residual components in such an order that we can ignore blockers in the solved residual components.

Recall that a charge $q(Q)$, for some $Q \subseteq F$, is the difference between the numbers of the start positions and the target positions in Q . Initially, if there is an edge $e \in E^H$ such that removing e splits H into two subtrees with zero total charge each, then we remove e from H and recurse on the two subtrees.

Let us now assign an orientation to the edges of H in the following way. For each edge $e = (u, v)$, let H_u and H_v be the two trees of $H \setminus \{e\}$ containing u and v , respectively. We orient e from u to v if $q(H_u) > 0 > q(H_v)$, and from v to u if $q(H_u) < 0 < q(H_v)$.

► **Lemma 10.** *There exists at least one sink vertex in the directed acyclic graph H .*

Using Lemma 10, our algorithm selects a sink node σ of H . The respective residual component \bar{F}_σ is solved as follows. First, all robots inside \bar{F}_σ are moved to unoccupied target positions. Since all incident edges of σ in H are directed inwards, each edge requires one or more robot(s) to move into \bar{F}_σ . The exact number can be computed from the charges of the subtrees of the adjacent residual components. We then move the required number of robots the adjacent residual components (and farther residual components if needed) to \bar{F}_σ over the corresponding blocking areas. Consider the blocker t associated with a blocking area b incident to \bar{F}_σ . If t is occupied before the charge of \bar{F}_σ becomes zero, then t has to reside in \bar{F}_σ , as the adjacent residual components were not yet processed by the algorithm. Then we move the robot from t to another unoccupied target in \bar{F}_σ , and the now unoccupied blocker position t is the last target to become occupied by a robot moving across b .

Once all target positions of \bar{F}_σ are filled, σ and its incident edges are removed from H , and we recurse on the remaining subtrees. Due to the way we select σ , and the number of robots that are moved into \bar{F}_σ , each subtree has a total zero-charge.

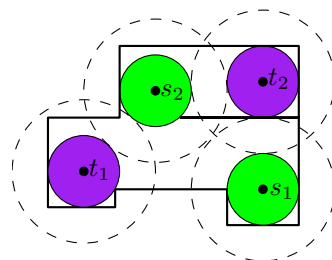
► **Theorem 11.** *When the free space consists of a single connected component, our algorithm finds a solution to the unlabeled motion planning problem for unit-disc robots in a simple workspace, assuming monochromatic separation $\mu = 4$ and bichromatic separation $\beta = 2$. This takes $O(n \log n + mn + m^2)$ time.*

We now show how to extend our approach to $\beta = 0$. In the above algorithm, we use the bichromatic separation $\beta = 2$ to ensure that at every moment in time any subset of nodes of the motion graph can be occupied by robots. If $\beta < 2$ we can no longer assume that any start position and any target position can be occupied at the same time. Nevertheless, even when $\beta = 0$, observe that, due to $\mu = 4$, for a pair of start and target positions s_i and t_j such that $|s_i t_j| < 2$, no other target position t_k can lie in $\mathcal{D}_2(s_i)$, and no other start position s_ℓ can lie in $\mathcal{D}_2(t_j)$. Thus, there is a guaranteed path from s_i to t_j . This can be exploited to adjust the motion graph and the algorithms for $\beta = 0$. Specifically, for each pair of such s_i and t_j , we create a single target node in our motion graph, we move the robot from s_i to t_j , and adjust our algorithm to work for the case of different number of start and target nodes in the motion graph.

► **Theorem 12.** *When the free space \mathcal{F} consists of a single component, the algorithm finds a solution to the unlabeled motion planning problem for unit-disc robots in a simple workspace, assuming monochromatic separation $\mu = 4$. This takes $O(n \log n + mn + m^2)$ time.*

5 Multiple free space components

In this section we consider the case where the free space \mathcal{F} consists of multiple connected components. Since a separation of $\beta = 3$ is necessary to guarantee a solution, we now assume separation bounds of $\mu = 4$ and $\beta = 3$.



■ **Figure 8** An example of a position (s_2) blocking movement (s_1 to t_1) in another free space component.

Within each free space component we can use the algorithm from Section 4. However, paths, that are otherwise valid, may be blocked by a robot from another component (see Figure 8). In this example, there is a simple solution: Move the robot away from s_2 toward t_2 in the upper component, before moving the robot from s_1 to t_1 in the lower component. In the following we prove that there always exists an order on the free space components such that the motion planning problem can be solved by solving the problem component by component in that order.

Let F, F' be two distinct components of \mathcal{F} , and let $x \in F$ be such that $\mathcal{D}_2(x) \cap F' \neq \emptyset$. Since the workspace is simple, it is sufficient to prove that for any such pair of components there is an order between them such that we can first solve one component and then the other. There are two reasons why such an ordering may not exist.

12:14 Unlabeled Multi-Robot Motion Planning with Tighter Separation Bounds

Firstly, there might be a start position $s \in F$ and a start position $s' \in F'$ such that $\mathcal{D}_2(s) \cap F' \neq \emptyset$ and $\mathcal{D}_2(s') \cap F \neq \emptyset$, that is, a start position in F interferes with paths in F' and vice versa. However, Adler et al. [1] proved that with $\mu = 4$, this cannot be the case. Likewise it cannot happen that a target position in F interferes with paths in F' and at the same time a target position in F' interferes with paths in F .

Secondly, there might be a start and target positions $s, t \in F$ both interfering with paths in F' . Because we only have a separation bound of $\beta = 3$ between s and t , this may actually occur. However, interference does not always affect the connectivity of the affected free space component. Therefore, we define a position x (start or target) to be a *remote blocker* of a free space component F' if (1) $x \notin F'$, and (2) $\mathcal{D}_2(x)$ intersects $\partial F'$ in more than one connected component.

► **Lemma 13.** *If the unlabeled motion planning problem has no remote blockers, then there is always a solution.*

Now the key geometric observation is that if auras of both s and t intersect F' , they cannot be both remote blockers of F' , and as a consequence we can still always find an order to resolve F and F' .

► **Theorem 14.** *We are given m unit-disc robots in a simple polygonal workspace $\mathcal{W} \subset \mathbb{R}^2$, with start and target positions S, T and separation constraints $\mu = 4$ and $\beta = 3$. Assuming each connected component F of the free space \mathcal{F} for a single unit-disc robot in \mathcal{W} contains an equal number of start and target positions, there exists a collision-free motion plan for the robots starting at S such that all target positions in T are occupied after execution.*

6 Conclusion

In this paper we presented an efficient algorithm for the unlabeled motion planning problem for unit-disc robots with sufficient separation in a simple polygon. Our result is optimal, in the sense that with less separation a solution may not exist. Nevertheless, there remain a number of challenging open problems.

To prove the tightness of the separation bounds, we first constructed domains with straight-line segments and circular arcs as boundaries, and then obtained simple polygons by approximating these. This results in polygons of high complexity. An open question remains whether it is possible to prove the separation bounds with constant-complexity polygons.

Of course, a solution may still exist even if the separation bounds are violated. The complexity of the problem in this setting remains a challenging open problem. The general unlabeled motion planning problem in a polygonal environment with holes is PSPACE-complete [2, 20]. Does the restriction to unit-disc robots and/or simple domains make the problem tractable, in particular if we still enforce some small separation bound?

What challenges arise when the workspace is no longer simple, but rather contains obstacles? Intuitively, obstacles seem to pose an issue when defining an ordering for solving multiple free space components, since positions can interfere between components at multiple locations. Are there conditions, similar to the separation bounds, which can guarantee a solution (together with an efficient algorithm) for unlabeled multi-robot motion planning amidst obstacles?

References

- 1 Aviv Adler, Mark de Berg, Dan Halperin, and Kiril Solovey. Efficient multi-robot motion planning for unlabeled discs in simple polygons. In *Algorithmic Foundations of Robotics XI*, pages 1–17. Springer, 2015.
- 2 Thomas Brocken, G. Wessel van der Heijden, Irina Kostitsyna, Lloyd E. Lo-Wong, and Remco J. A. Surtel. Multi-robot motion planning of k -colored discs is PSPACE-hard. In *10th International Conference on Fun with Algorithms (FUN 2021)*, volume 157 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:16, 2020.
- 3 Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, 2005.
- 4 Erik D. Demaine, Sándor P. Fekete, Phillip Keldenich, Henk Meijer, and Christian Scheffer. Coordinated motion planning: Reconfiguring a swarm of labeled robots with bounded stretch. *SIAM Journal on Computing*, 48(6):1727–1762, 2019.
- 5 Dan Halperin, Lydia Kavraki, and Kiril Solovey. Robotics. In Jacob E. Goodman, Joseph O’Rourke, and Csaba Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 51, pages 1343–1376. Chapman & Hall/CRC, 3rd edition, 2018.
- 6 Dan Halperin, Micha Sharir, and Oren Salzman. Algorithmic motion planning. In Jacob E. Goodman, Joseph O’Rourke, and Csaba Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 50, pages 1311–1342. Chapman & Hall/CRC, 3rd edition, 2018.
- 7 Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science*, 343:72–96, 2005.
- 8 John E. Hopcroft, Jacob Theodore Schwartz, and Micha Sharir. On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the “warehouseman’s problem”. *The International Journal of Robotics Research*, 3(4):76–88, 1984.
- 9 Lydia E. Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.
- 10 Stephen Kloder and Seth Hutchinson. Path planning for permutation-invariant multirobot formations. *IEEE Transactions on Robotics*, 22(4):650–665, 2006.
- 11 Daniel M. Kornhauser, Gary Miller, and Paul Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. Master’s thesis, MIT, Dept. of Electrical Engineering and Computer Science, 1984.
- 12 James J. Kuffner and Steven M. Lavalle. RRT-Connect: An efficient approach to single-query path planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 995–1001, 2000.
- 13 Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- 14 Gildardo Sanchez and Jean-Claude Latombe. Using a PRM planner to compare centralized and decoupled planning for multi-robot systems. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 2112–2119. IEEE, 2002.
- 15 Jacob T. Schwartz and Micha Sharir. On the “piano movers” problem. II. General techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics*, 4(3):298–351, 1983.
- 16 Jacob T. Schwartz and Micha Sharir. On the piano movers’ problem: III. coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers. *The International Journal of Robotics Research*, 2(3):46–75, 1983.
- 17 Micha Sharir and Shmuel Sifrony. Coordinated motion planning for two independent robots. *Annals of Mathematics and Artificial Intelligence*, 3(1):107–130, 1991.
- 18 Rahul Shome, Kiril Solovey, Andrew Dobson, Dan Halperin, and Kostas E. Bekris. dRRT^{*}: Scalable and informed asymptotically-optimal multi-robot motion planning. *Autonomous Robots*, 44(3-4):443–467, 2020.

- 19 Israela Solomon and Dan Halperin. Motion planning for multiple unit-ball robots in \mathbb{R}^d . In Marco Morales, Lydia Tapia, Gildardo Sánchez-Ante, and Seth Hutchinson, editors, *Proc. 13th Workshop on the Algorithmic Foundations of Robotics, WAFR*, volume 14 of *Springer Proceedings in Advanced Robotics*, pages 799–816. Springer, 2018.
- 20 Kiril Solovey and Dan Halperin. On the hardness of unlabeled multi-robot motion planning. *The International Journal of Robotics Research*, 35(14):1750–1759, 2016.
- 21 Kiril Solovey, Oren Salzman, and Dan Halperin. Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning. *International Journal of Robotics Research*, 35(5):501–513, 2016.
- 22 Kiril Solovey, Jingjin Yu, Or Zamir, and Dan Halperin. Motion planning for unlabeled discs with optimality guarantees. In *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, 2015.
- 23 Paul Spirakis and Chee K. Yap. Strong NP-hardness of moving many discs. *Information Processing Letters*, 19(1):55–59, 1984.
- 24 Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Roman Barták, and Eli Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In Pavel Surynek and William Yeoh, editors, *Proc. 12th International Symposium on Combinatorial Search, SOCS*, pages 151–159. AAAI Press, 2019.
- 25 Petr Svestka and Mark H. Overmars. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, 23(3):125–152, 1998.
- 26 Matthew Turpin, Nathan Michael, and Vijay Kumar. Concurrent assignment and planning of trajectories for large teams of interchangeable robots. In *IEEE International Conference on Robotics and Automation*, pages 842–848. IEEE, 2013.
- 27 Glenn Wagner and Howie Choset. Subdimensional expansion for multirobot path planning. *Artificial Intelligence*, 219:1–24, 2015.
- 28 Chee Yap. Coordinating the motion of several discs. *Courant Institute of Mathematical Sciences*, 1984.
- 29 Jingjin Yu. Constant factor time optimal multi-robot routing on high-dimensional grids. In Hadas Kress-Gazit, Siddhartha S. Srinivasa, Tom Howard, and Nikolay Atanasov, editors, *Robotics: Science and Systems XIV*, 2018.
- 30 Jingjin Yu and Steven M. LaValle. Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. *IEEE Transactions on Robotics*, 32(5):1163–1177, 2016.

Optimality of the Johnson-Lindenstrauss Dimensionality Reduction for Practical Measures

Yair Bartal ✉

Hebrew University, Jerusalem, Israel

Ora Nova Fandina ✉

Aarhus University, Denmark

Kasper Green Larsen ✉

Aarhus University, Denmark

Abstract

It is well known that the Johnson-Lindenstrauss dimensionality reduction method is optimal for worst case distortion. While in practice many other methods and heuristics are used, not much is known in terms of bounds on their performance. The question of whether the JL method is optimal for practical measures of distortion was recently raised in [8] (NeurIPS'19). They provided upper bounds on its quality for a wide range of practical measures and showed that indeed these are best possible in many cases. Yet, some of the most important cases, including the fundamental case of average distortion were left open. In particular, they show that the JL transform has $1 + \epsilon$ average distortion for embedding into k -dimensional Euclidean space, where $k = O(1/\epsilon^2)$, and for more general q -norms of distortion, $k = O(\max\{1/\epsilon^2, q/\epsilon\})$, whereas tight lower bounds were established only for large values of q via reduction to the worst case.

In this paper we prove that these bounds are best possible for any dimensionality reduction method, for any $1 \leq q \leq O(\frac{\log(2\epsilon^2 n)}{\epsilon})$ and $\epsilon \geq \frac{1}{\sqrt{n}}$, where n is the size of the subset of Euclidean space.

Our results also imply that the JL method is optimal for various distortion measures commonly used in practice, such as *stress*, *energy* and *relative error*. We prove that if any of these measures is bounded by ϵ then $k = \Omega(1/\epsilon^2)$, for any $\epsilon \geq \frac{1}{\sqrt{n}}$, matching the upper bounds of [8] and extending their tightness results for the full range moment analysis.

Our results may indicate that the JL dimensionality reduction method should be considered more often in practical applications, and the bounds we provide for its quality should be served as a measure for comparison when evaluating the performance of other methods and heuristics.

2012 ACM Subject Classification Theory of computation \rightarrow Random projections and metric embeddings; Theory of computation \rightarrow Computational geometry; Theory of computation \rightarrow Unsupervised learning and clustering

Keywords and phrases average distortion, practical dimensionality reduction, JL transform

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.13

Related Version *Full Version*: <https://arxiv.org/abs/2107.06626>

Funding *Yair Bartal*: Supported in part by a grant from the Israeli Science Foundation (1817/17). *Kasper Green Larsen*: Supported by Independent Research Fund Denmark (DFR) Sapere Aude Research Leader grant No 9064-00068B.

1 Introduction

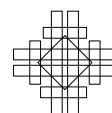
Dimensionality reduction is a key tool in numerous fields of data analysis, commonly used as a compression scheme to enable reliable and efficient computation. In metric dimensionality reduction subsets of high-dimensional spaces are embedded into a low-dimensional space, attempting to preserve metric structure of the input. There is a large body of theoretical and applied research on such methods spanning a wide range of application areas such as online algorithms, computer vision, network design, machine learning, to name a few.



© Yair Bartal, Ora Nova Fandina, and Kasper Green Larsen;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 13; pp. 13:1–13:16
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Metric embedding has been extensively studied by mathematicians and computer scientists over the past few decades (see [18, 25, 19] for surveys). developing a rich theory, and some original and elegant techniques have been developed and successfully applied in various fields of algorithmic research. See [27, 18, 34] for exposition of some applications.

The vast majority of these methods have been designed to optimize the worst-case distance error incurred by embedding. For metric spaces (X, d_X) and (Y, d_Y) an injective map $f : X \rightarrow Y$ is an embedding. It has (a worst-case) distortion $\alpha \geq 1$ if there is a positive constant c satisfying for all $u \neq v \in X$, $d_Y(f(u), f(v)) \leq c \cdot d_X(u, v) \leq \alpha \cdot d_Y(f(u), f(v))$. A cornerstone result in metric dimensionality reduction is the celebrated Johnson-Lindenstrauss lemma [21]. It states that any n -point subset of Euclidean space can be embedded, via a linear transform, into a $O(\log n/\epsilon^2)$ -dimensional subspace with $1 + \epsilon$ distortion. It has been recently shown to be optimal in [24] and in [6] (improving upon [5]). Furthermore, it was shown in [26] that there are Euclidean pointsets in \mathbb{R}^d for which any embedding into k -dimensions must have $n^{\Omega(1/k)}$ distortion, effectively ruling out dimensionality reduction into a constant number of dimensions with a constant worst-case distortion.

Metric embedding and in particular dimensionality reduction have also gained significant attention in applied community. Practitioners have frequently employed classic tools of metric embedding as well as have designed new techniques to cope with high-dimensional data. A large number of dimensionality reduction heuristics have been developed for a variety of practical settings, eg. [33, 28, 7, 36]. However, most of these heuristics have not been rigorously analyzed in terms of the incurred error. Recent papers [11] and [8] initiate the formal study of practically oriented analysis of metric embedding.

Practical distortion measures. In contrast to the worst case distortion the quality of practically motivated embedding is often determined by its average performance over all pairs, where an error per pair is measured as an additive error, a multiplicative error or a combination of both. There is a huge body of applied research investigating such notions of quality. For the list of citations and a more detailed account on the theoretical and practical importance of average distortion measures see [8].

In this paper we consider the most basic and commonly used in practical applications notions of average distortion, which we define in the following. Moment of distortion was defined in [4], and studied in various papers since then.

► **Definition 1** (ℓ_q -distortion). For $u \neq v \in X$ let $expans_f(u, v) = d_Y(f(u), f(v))/d_X(u, v)$ and $contract_f(u, v) = d_X(u, v)/d_Y(f(u), f(v))$. Let $dist_f(u, v) = \max\{expans_f(u, v), contract_f(u, v)\}$. For any $q \geq 1$ the q -th moment of distortion is defined by

$$\ell_q\text{-dist}(f) = \left(\frac{1}{\binom{|X|}{2}} \sum_{u \neq v \in X} (dist_f(u, v))^q \right)^{1/q}.$$

Additive average distortion measures are often used when a nearly isometric embedding is expected. Such notions as *energy*, *stress* and *relative error measure* (REM) are common in various statistic related applications. For a map $f : X \rightarrow Y$, for a pair $u \neq v \in X$ let $d_{u,v} := d_X(u, v)$ and let $\hat{d}_{uv} := d_Y(f(u), f(v))$. For all $q \geq 1$

► **Definition 2** (Additive measures).

$$Energy_q(f) = \left(\frac{1}{\binom{|X|}{2}} \sum_{u \neq v \in X} \left(\frac{|\hat{d}_{uv} - d_{u,v}|}{d_{u,v}} \right)^q \right)^{\frac{1}{q}} = \left(\frac{1}{\binom{|X|}{2}} \sum_{u \neq v \in X} |expans_f(u, v) - 1|^q \right)^{\frac{1}{q}}.$$

$$Stress_q(f) = \left(\frac{\sum_{u \neq v \in X} |\hat{d}_{uv} - d_{uv}|^q}{\sum_{u \neq v \in X} (d_{uv})^q} \right)^{\frac{1}{q}}, \quad Stress^*_q(f) = \left(\frac{\sum_{u \neq v \in X} |\hat{d}_{uv} - d_{uv}|^q}{\sum_{u \neq v \in X} (\hat{d}_{uv})^q} \right)^{\frac{1}{q}}.$$

$$REM_q(f) = \left(\frac{1}{\binom{|X|}{2}} \sum_{u \neq v \in X} \left(\frac{|\hat{d}_{uv} - d_{uv}|}{\min\{\hat{d}_{uv}, d_{uv}\}} \right)^q \right)^{\frac{1}{q}}.$$

▷ Claim 3 ([8]). For all $q \geq 1$, $\ell_q\text{-dist}(f) - 1 \geq REM_q(f) \geq Energy_q(f)$.

In the full version we also address the machine learning motivated σ -distortion [12].

In [8] the authors rigorously analyzed dimensionality reduction for the above distortion measures. The central question they studied is: *What dimensionality reduction method is optimal for these quality measures and what are the optimal bounds achievable? In particular, is the Johnson-Lindenstrauss (JL) transform also optimal for the average quality criteria?*

Their analysis of the Gaussian implementation of the JL embedding [20] shows that any Euclidean subset can be embedded with $1 + \epsilon$ average distortion ($\ell_1\text{-dist}$) into $k = O(1/\epsilon^2)$ dimensions. And for more general case of the q -moment of distortion, the dimension is $k = O(\max\{1/\epsilon^2, q/\epsilon\})$. However, tight lower bounds were proved only for large values of q , leaving the question of optimality of the most important case of small q , and particularly the most basic case of $q = 1$, unresolved.

For the additive average measures (stress, energy and others) they prove that a bound of ϵ can be achieved in dimension $k = O(q/\epsilon^2)$. They showed a tight lower bound on the required dimension only for $q \geq 2$, leaving the basic case of $q = 1$ also unresolved.

In this paper, we prove that indeed the Johnson-Lindenstrauss bounds are best possible for any dimensionality reduction for the full range of $q \geq 1$, for all the average distortion measures defined in this paper. We believe that besides theoretical contribution this statement may have important implications for practical considerations. In particular, it may affect the way the JL method is viewed and used in practice, and the bounds we give may serve a basis for comparison for other methods and heuristics.

Our results. We show that the guarantees given by the Gaussian JL implementation are the best possible for the average distortion measures. In particular, we prove

► **Theorem 4.** *Given any integer n and $\Omega(\frac{1}{\sqrt{n}}) < \epsilon < 1$, there exists a $\Theta(n)$ -point subset of Euclidean space such that any map f of it into ℓ_2^k with $\ell_1\text{-dist}(f) \leq 1 + \epsilon$ requires $k = \Omega(1/\epsilon^2)$.*

► **Theorem 5.** *Given any integer n , and $\Omega(\frac{1}{\sqrt{n}}) < \epsilon < 1$, and $1 \leq q \leq O(\log(\epsilon^2 n)/\epsilon)$, there exists a $\Theta(n)$ -point subset of Euclidean space such that any embedding of it into ℓ_2^k with ℓ_q -distortion at most $1 + \epsilon$ requires $k = \Omega(q/\epsilon)$.*

As ℓ_q -distortion is monotonically increasing as a function of q , the theorems imply the lower bound of $k = \Omega(\max\{1/\epsilon^2, q/\epsilon\})$. For the additive distortion measures we prove:

► **Theorem 6.** *Given any integer n and $\Omega(\frac{1}{\sqrt{n}}) < \epsilon < 1$, there exists a $\Theta(n)$ -point subset of Euclidean space such that any embedding of it into ℓ_2^k with any of $Energy_1$, $Stress_1$, $Stress^*_1$, REM_1 or σ -distortion bounded above by ϵ requires $k = \Omega(1/\epsilon^2)$.*

Our main proof is of the lower bound for $Energy_1$ measure, which we show to imply the bound in Theorem 4 and for all measures in Theorem 6, with some small modifications for the stress measures. Furthermore, since all additive measures are monotonically increasing with

13:4 Optimality of the JL Dimensionality Reduction for Practical Measures

q the bounds hold for all $q \geq 1$. Therefore Theorems 4 and 5 together provide a tight bound of $\Omega(\max\{1/\epsilon^2, q/\epsilon\})$ for the ℓ_q -distortion. Additionally combined with the lower bounds of [8] for $q \geq 2$, Theorem 6 provides a tight bound of $\Omega(q/\epsilon^2)$ for all additive measures.

Techniques. The proofs of the lower bounds in all the theorems are based on counting argument, as in the lower bound for the worst case distortion proven by [24]. We extend the framework of [24] to the entire range of q moments of distortion, including the average distortion. As in the original proof we show that there exists a large family \mathcal{P} of metric spaces that are quite different from each other so that if one can embed all of these into a Euclidean space with a small average distortion the resulting image spaces are different too. This implies that if the target dimension k is too small there is not enough space to accommodate all the different metric spaces from the family.

Let us first describe the framework of [24].¹ The main idea is to construct a large family of n -point subspaces $I \subseteq \ell_2^{\Theta(n)}$ so that each subspace in the family can be uniquely encoded using a small number of bits, assuming that each I can be embedded with $1 + \epsilon$ worst-case distortion in ℓ_2^k . The sets they construct are such that the information on the inner products between all the points in I , even if distorted by an additive error of $O(\epsilon)$, enables full reconstruction of the points in the set. In particular, each I consists of a zero vector together with the standard basis vectors E and an additional set of vectors denoted by Y . The set Y is defined in such a way that $\langle y, e \rangle \in \{0, c\epsilon\}$, for a constant $c > 1$, for all $(y, e) \in Y \times E$. The authors then show that a $1 + \epsilon$ distortion embedding f of I must map all the points into the ball of radius 2 while preserving all the inner products up to an additive error $\Theta(\epsilon)$, which enables to recover the vectors in Y . The next step is to show that all image points can be encoded using a small number of bits, while preserving the inner product information up to an $\Theta(\epsilon)$ additive error. This can be achieved by carefully discretizing the ball, and applying a map \tilde{f} mapping every point to its discrete image approximation so that $\langle f(v), f(u) \rangle = \langle \tilde{f}(v), \tilde{f}(u) \rangle \pm \Theta(\epsilon)$. For this purpose one may use the method of [6] who showed² that randomly rounding the image points to the points in a small enough grid will preserve all the pairwise inner products within $\Theta(\epsilon)$ additive error with constant probability, and this in turn allows to derive a short binary encoding for each input point. This implies the lower bound on $k = \Omega(\log(\epsilon^2 n)/\epsilon^2)$, for $\epsilon = \Omega(1/\sqrt{n})$.

Let us now explain the challenges in applying this method to the case of bounded average distortion and q -moments. Assuming $f : I \rightarrow \ell_2^k$ has $1 + \epsilon$ average distortion neither implies that all images are in a ball of constant radius nor that f preserves all pairwise inner products. The bounded average distortion also does not guarantee the existence of a large subset of I with the properties above. We suggest the following approach to overcoming these issues. First, we add to I a large number of "copies" of 0 vectors which enables to argue that a large subset $\hat{I} \subseteq I$ will be mapped into a constant radius ball, such that the average additive distortion is $\Theta(\epsilon)$. The next difficulty is that if the images would be rounded to the points in a grid using a mapping which would preserve *all* pairwise inner products with $\Theta(\epsilon)$ additive error, then the resulting grid would be too large, which would't allow a sufficiently short encoding. We therefore round the images to a grid with $\Theta(\epsilon)$ additive approximation to the *average* of the inner products of \hat{I} and thus reduce the size of the grid (and the encoding). The next step is showing that the above guarantees imply the existence of a large enough

¹ The description is based on combining the methods of [24, 6], and can be also viewed as our q -moments bound with $q = \Theta(\log(\epsilon^2 n)/\epsilon)$.

² The original proof of [24] uses a different elegant discretization argument.

subset of pairs $\mathcal{Z} \subseteq \binom{I}{2}$ of special structure, which allows to encode the *entire* set I with a few bits even if only the partial information about the inner products within \mathcal{Z} is approximately preserved. In particular, we show that there is a large subset $\mathcal{Y}^G \subseteq Y$ such that for each point $y \in \mathcal{Y}^G$ there is a large enough subset $\mathcal{E}_y \subseteq E$ such that all pairwise inner products $\langle y, e \rangle$, where $y \in \mathcal{Y}^G$ and $e \in \mathcal{E}_y$, are additively preserved up to $\Theta(\epsilon)$ in the grid embedding, and therefore all the discretized images of these points have short binary encoding. The last step is to argue that this subset is sufficiently large so the knowledge of its approximate inner products possesses enough information in order to recover the entire point set I from our small size encoding. As this set still covers only a constant fraction of the pairs, and encoding the rest of the points is far more costly, this forces the dimension and number of points in our instance to be set to $d = \Theta(n) = \Theta(1/\epsilon^2)$, implying a lower bound of $k = \Omega(1/\epsilon^2)$. Finally, we prove that this can extend to arbitrary large subspaces via metric composition techniques. To extend these ideas to the general case of q -moments of distortion we prove that similar additive approximation distortion bounds hold with high probability of at least $1 - e^{-\Theta(\epsilon q)}$. This means that a smaller fraction of the pairs require a more costly encoding, and allows us to set $d = \Theta(n) = \Theta(1/\epsilon^2) \cdot e^{\Theta(\epsilon q)}$, implying a lower bound of $k = \Omega(q/\epsilon)$.

Related work. The study of "beyond the worst-case" distortion analysis of metric embedding initiated in [22] by introducing partial and scaling distortions. This has generated a rich line of follow up work, [1, 4, 2] just to name a few. The notions of average distortion and ℓ_q -distortion were introduced in [4] who gave bounds on embedding general metrics in normed spaces. Other notions of refined distortion analysis considered in the literature include such notions as Ramsey type embeddings [9], local distortion embeddings [3], terminal and prioritized distortion [15, 14], and recent works on distortion of the q -moments³[29, 30, 23].

In applied community, various notions of average distortion are frequently used to infer the quality of heuristic methods [17, 16, 32, 13, 31, 35, 10].

However, the only work rigorously analyzing these notions we are aware of is that of [8]. They proved lower bounds of $k = \Omega(1/\epsilon)$ for the all additive measures average (1-norm) version, and for the average distortion measure (ℓ_1 -distortion), which we improve here to the tight $\Omega(1/\epsilon^2)$ bound. For $q \geq 2$ they gave tight bounds of $\Omega(q/\epsilon^2)$ for all additive measures. For ℓ_q -*dist* they have shown that for $q = \Omega(\log(1/\epsilon)/\epsilon)$ the tight bound of $k = \Omega(q/\epsilon)$ follows from the black-box reduction to the lower bound on the worst case distortion.

2 Lower bound for average distortion and additive measures

In this section we prove Theorems 4 and Theorem 6. Using Claim 3, we may focus on proving the lower bound for $Energy_1(f)$ in order to obtain similar lower bounds for $REM_1(f)$ and ℓ_1 -*dist*(f). In full version of the paper we show how to change this proof in order to obtain lower bound on $Stress_1(f)$, and further show that the lower bounds for all the additive measures follow from the lower bounds on Energy and Stress.

We present here the proof of an existence of a bad metric space of size $\hat{n} = \Theta(1/\epsilon^2)$, while construction of a metric space of an arbitrary size $n \geq \hat{n}$, based on a similar technique appearing in [8] via metric composition [9], is given in the full version of the paper.

³ The notion in these papers, also studied [4, 8], computes the ratio between the average of (or q -moments) of new distances to that of original distances, in contrast to the average distortion (or q -moments of distortion) measure in Definition 1, which measures the average (or q -moments) of pairwise distortions.

13:6 Optimality of the JL Dimensionality Reduction for Practical Measures

We construct a large family \mathcal{P} of metric spaces, such that each $I \in \mathcal{P}$ can be completely recovered by computing the inner products between the points in I . For a given $\epsilon < 0$, let $l = \lceil \frac{1}{\gamma^2 \epsilon^2} \rceil$, for some large constant $\gamma > 1$ to be determined later. We will prove $k \geq \frac{c}{\gamma^2} \frac{1}{\epsilon^2}$, for $c < 1$, and so we may assume w.l.o.g. that $\epsilon \leq 1/\gamma$, otherwise the statement trivially holds. We construct point sets $I \subset \ell_2^d$, where $d = 2l$, each I of size $3d = 6l = \Theta(1/\epsilon^2)$.

Define a set $O = \{o_j\}_{j=1}^d$ of d arbitrary near zero vectors in ℓ_2^d , i.e., a set of d different vectors such that for all $o_j \in O$, $\|o_j\|_2 \leq \epsilon/100$. Let $E = \{e_1, e_2, \dots, e_d\}$ denote the vectors of the standard basis of \mathbb{R}^d . For a set S of l indices from $\{1, 2, \dots, d\}$, we define $y_S = \frac{1}{\sqrt{l}} \sum_{j \in S} e_j$. For a sequence of d index sets (possibly with repetitions) S_1, S_2, \dots, S_d , let $Y[S_1, \dots, S_d] = \{y_{S_1}, \dots, y_{S_d}\}$. Each point set $I[S_1, \dots, S_d] \in \mathcal{P}$ is defined as the union of the sets defined above⁴, i.e., $I[S_1, \dots, S_d] = O \cup E \cup Y[S_1, \dots, S_d]$. The size of the family is $|\mathcal{P}| = \binom{d}{l}^d$. Note that each $I \in \mathcal{P}$ is contained in $B_2(1)$, the unit ball of ℓ_2^d , and has diameter $\text{diam}(I) = \sqrt{2}$. Additionally, for all $e_j \in E$ and $y_S \in Y$ the value of the inner product $\langle e_j, y_S \rangle$ determines whether $e_j \in \text{span}\{e_i | i \in S\}$. In particular, if $\langle e_j, y_S \rangle = 0$ then $j \notin S$, and if $\langle e_j, y_S \rangle = 1/\sqrt{l} \geq (1/2)\gamma\epsilon$ then $j \in S$.

Assume that for each $I \in \mathcal{P}$ there is an embedding $f : I \rightarrow \ell_2^k$, with $\text{Energy}_1(f) \leq \epsilon$. We prove that this implies that $k = \Omega(1/\epsilon^2)$. The strategy is to produce a unique binary encoding of each I in the family based on the embedding f . Let $\text{length}(I)$ denote the length of the encoding for each I , we will show that $\text{length}(I) \lesssim l^2 + l \cdot k \log(\frac{1}{\epsilon k})$. Since the encoding defines an injective map from \mathcal{P} to $\{0, 1\}^{\text{length}(I)}$, the number of different sets that can be recovered by decoding is at most $2^{\text{length}(I)}$. Now, because $|\mathcal{P}| = \binom{d}{l}^d \geq 2^{2l^2}$ we get that $k \log(\frac{1}{\epsilon k}) \gtrsim l$ and show that this implies the bound on $k \geq \Omega(l)$.

We are now set to describe the encoding for each I and to bound its length. First, in the following lemma, we show that there exists a large subset $\hat{I} \subseteq I$ that is mapped by f into a ball of a constant radius in k -dimensional space and that the average of the errors in the inner products incurred by f on the subset \hat{I} is bounded by $\Theta(\epsilon)$.

► **Lemma 7.** *For any $I \in \mathcal{P}$ let $f : I \rightarrow \ell_2^k$ be an embedding with $\text{Energy}_1(f) \leq \epsilon$, with $\epsilon \leq 1/36$. Let $0 < \alpha \leq 1/16$ be a parameter. There is a subset $\hat{I} \subseteq I$ of size $|\hat{I}| \geq (1 - \alpha)|I|$ such that $f(\hat{I}) \subset B_2(1 + \frac{3.01\epsilon}{\alpha})$, and $\frac{1}{\binom{I}{2}} \sum_{(u,v) \in \binom{I}{2}} |\langle f(u), f(v) \rangle - \langle u, v \rangle| \leq (10 + \frac{1}{2\alpha})\epsilon$.*

Proof. By assumption we have that the following condition holds:

$$\text{Energy}_1(f) = \frac{1}{\binom{I}{2}} \sum_{(u,v) \in \binom{I}{2}} |\text{expans}_f(u, v) - 1| \leq \epsilon. \quad (1)$$

This bound implies that

$$\begin{aligned} \frac{1}{|O|(|I| - 1)} \sum_{o_j \in O} \sum_{v \in I, v \neq o_j} |\text{expans}_f(o_j, v) - 1| &\leq \frac{1}{|O|(|I| - 1)} \sum_{u \neq v \in I} |\text{expans}_f(u, v) - 1| \\ &\leq \frac{3d(3d - 1)}{d(3d - 1)} \epsilon = 3\epsilon. \end{aligned}$$

From which follows that

$$\min_{o_j \in O} \frac{1}{|I| - 1} \sum_{v \in I, v \neq o_j} |\text{expans}_f(o_j, v) - 1| \leq 3\epsilon. \quad (2)$$

⁴ We will omit $[S_1, \dots, S_d]$ from notation for a fixed choice of the sets.

Let $\hat{o} \in O$ denote the point at which the minimum is obtained. We assume without loss of generality that $f(\hat{o}) = 0$. Let \hat{I} be the set of all $v \in I$ such that $|expans_f(\hat{o}, v) - 1| \leq \frac{3\epsilon}{\alpha}$. By Markov's inequality, $|\hat{I}| \geq (1 - \alpha)|I|$. We have that for all $v \in \hat{I}$, $|expans_f(v, \hat{o}) - 1| = \left| \frac{\|f(v)\|_2}{\|v - \hat{o}\|_2} - 1 \right| \leq \frac{3\epsilon}{\alpha}$, and using $\|v - \hat{o}\|_2 \leq \|v\|_2 + \|\hat{o}\|_2 \leq 1 + \epsilon/100$, so that $\|f(v)\|_2 \leq (1 + \frac{3\epsilon}{\alpha})(1 + \epsilon/100) \leq 1 + \frac{3.002\epsilon}{\alpha}$, implying that $f(v) \in B_2(1 + \frac{3.01\epsilon}{\alpha})$.

For all $(u, v) \in \binom{\hat{I}}{2}$ we have:

$$\begin{aligned} |\langle f(u), f(v) \rangle - \langle u, v \rangle| &\leq \frac{1}{2} \left[\left| \|f(u)\|_2^2 - \|u\|_2^2 \right| + \left| \|f(v)\|_2^2 - \|v\|_2^2 \right| \right] \\ &\quad + \frac{1}{2} \left[\left| \|f(u) - f(v)\|_2^2 - \|u - v\|_2^2 \right| \right]. \end{aligned}$$

We can bound each term as follows:

$$\begin{aligned} \left| \|f(u)\|_2^2 - \|u\|_2^2 \right| &= \left| \|f(u) - f(\hat{o})\|_2^2 - \|u - \hat{o}\|_2^2 + \|u - \hat{o}\|_2^2 - \|u\|_2^2 \right| \\ &\leq \left| \|f(u) - f(\hat{o})\|_2 - \|u - \hat{o}\|_2 \right| \cdot (\|f(u) - f(\hat{o})\|_2 + \|u - \hat{o}\|_2) \\ &\quad + \left| \|u - \hat{o}\|_2 - \|u\|_2 \right| \cdot (\|u - \hat{o}\|_2 + \|u\|_2) \\ &\leq \|u - \hat{o}\|_2 \cdot |expans_f(u, \hat{o}) - 1| \cdot (\|f(u)\|_2 + \|u - \hat{o}\|_2) + \|\hat{o}\|_2 \cdot (\|u - \hat{o}\|_2 + \|u\|_2) \\ &\leq \left(1 + \frac{\epsilon}{100} \right) |expans_f(u, \hat{o}) - 1| \left(1 + \frac{3.002\epsilon}{\alpha} + 1 + \frac{\epsilon}{100} \right) + \frac{\epsilon}{100} \cdot \left(2 + \frac{\epsilon}{100} \right) \\ &\leq \left(2 + \frac{3.01\epsilon}{\alpha} \right) |expans_f(u, \hat{o}) - 1| + \frac{\epsilon}{40} \leq \left(2 + \frac{1}{9\alpha} \right) |expans_f(u, \hat{o}) - 1| + \frac{\epsilon}{40}, \end{aligned}$$

where we have used $\|\hat{o}\|_2 \leq \epsilon/100$, $\|u - \hat{o}\|_2 \leq \|u\|_2 + \|\hat{o}\|_2 \leq 1 + \epsilon/100$, and the bound on the norms of the embedding within \hat{I} . Additionally, we have that

$$\begin{aligned} \left| \|f(u) - f(v)\|_2^2 - \|u - v\|_2^2 \right| &= \left| \|f(u) - f(v)\|_2 - \|u - v\|_2 \right| (\|f(u) - f(v)\|_2 + \|u - v\|_2) \\ &\leq \|u - v\|_2 |expans_f(u, v) - 1| (\|f(u)\|_2 + \|f(v)\|_2 + \|u - v\|_2) \\ &\leq \sqrt{2} \left(2 \left(1 + \frac{3.002\epsilon}{\alpha} \right) + \sqrt{2} \right) |expans_f(u, v) - 1| \leq \left(5 + \frac{1}{4\alpha} \right) |expans_f(u, v) - 1|, \end{aligned}$$

where the second to last inequality holds since $\|u - v\|_2 \leq diam(I) = \sqrt{2}$. It follows that:

$$\begin{aligned} \frac{1}{\binom{\hat{I}}{2}} \sum_{(u,v) \in \binom{\hat{I}}{2}} |\langle f(u), f(v) \rangle - \langle u, v \rangle| &\leq \tag{3} \\ &\leq \left(2 + \frac{1}{9\alpha} \right) \cdot \frac{1}{\binom{\hat{I}}{2}} \left(\frac{|\hat{I}| - 1}{2} \right) \sum_{u \in \hat{I}, u \neq \hat{o}} |expans_f(u, \hat{o}) - 1| \\ &\quad + \frac{1}{2} \left(5 + \frac{1}{4\alpha} \right) \cdot \frac{1}{\binom{\hat{I}}{2}} \sum_{(u,v) \in \binom{\hat{I}}{2}} |expans_f(u, v) - 1| + \frac{\epsilon}{40}. \end{aligned}$$

By definition of \hat{I} , and using (2) we have that

$$\begin{aligned} \frac{1}{\binom{\hat{I}}{2}} \left(\frac{|\hat{I}| - 1}{2} \right) \sum_{u \in \hat{I}, u \neq \hat{o}} |expans_f(u, \hat{o}) - 1| &= \frac{1}{|\hat{I}|} \sum_{u \in \hat{I}, u \neq \hat{o}} |expans_f(u, \hat{o}) - 1| \\ &\leq \frac{1}{|\hat{I}|} \sum_{u \in \hat{I}, u \neq \hat{o}} |expans_f(u, \hat{o}) - 1| \leq 3\epsilon. \end{aligned}$$

Therefore (3) yields that

$$\begin{aligned}
 & \frac{1}{|\hat{I}|} \sum_{(u,v) \in \binom{\hat{I}}{2}} |\langle f(u), f(v) \rangle - \langle u, v \rangle| \leq \\
 & \leq \left(2 + \frac{1}{9\alpha}\right) \cdot 3\epsilon + \frac{1}{2} \left(5 + \frac{1}{4\alpha}\right) \cdot \frac{1}{|\hat{I}|} \sum_{(u,v) \in \binom{\hat{I}}{2}} |\text{expans}_f(u, v) - 1| + \frac{\epsilon}{40} \\
 & \leq \frac{1}{2} \left(5 + \frac{1}{4\alpha}\right) \cdot \frac{1}{|\hat{I}|} \sum_{(u,v) \in \binom{\hat{I}}{2}} |\text{expans}_f(u, v) - 1| + \left(7 + \frac{1}{3\alpha}\right) \epsilon.
 \end{aligned}$$

Now, we have that

$$\frac{1}{|\hat{I}|} \sum_{(u,v) \in \binom{\hat{I}}{2}} |(\text{expans}_f(u, v)) - 1| \leq \frac{6}{5} \frac{1}{|\hat{I}|} \sum_{(u,v) \in \binom{\hat{I}}{2}} |(\text{expans}_f(u, v)) - 1| \leq \frac{6}{5} \epsilon,$$

using $|\hat{I}| \geq (1 - \alpha)|I|$, so that $\alpha \leq 1/16$ we have $|\hat{I}| \geq (1 - \frac{1}{3(1-\alpha)d})(1 - \alpha)^2 \cdot |I| \geq \frac{5}{8}|I|$ and applying (1). Finally, we obtain

$$\frac{1}{|\hat{I}|} \sum_{(u,v) \in \binom{\hat{I}}{2}} |\langle f(u), f(v) \rangle - \langle u, v \rangle| \leq \frac{6}{5} \cdot \frac{1}{2} \left(5 + \frac{1}{4\alpha}\right) \epsilon + \left(7 + \frac{1}{3\alpha}\right) \epsilon \leq \left(10 + \frac{1}{2\alpha}\right) \epsilon. \blacktriangleleft$$

We have shown thus far that for the large subset \hat{I} of the set I , the average of the inner products between the images equals up to an additive factor $\Theta(\epsilon)$ to the average of the inner products between the original points. Moreover, all the images of \hat{I} are in the constant radius ball. We next show that rounding these images to the (randomly chosen) points of the sufficiently small grid will not change the sum of the inner products too much, implying that instead of encoding the original images $f(\hat{I})$ we can encode its rounded counterpart. To show this, we use a technique of randomized rounding as proposed in [6].

► **Lemma 8.** *Let $X \subset \ell_2^k$ such that $X \subset B_2(r)$. For $\delta < r/\sqrt{k}$ let $G_\delta \subseteq B_2(r)$ denote the intersection of the δ -grid with $B_2(r)$. There is a mapping $g : X \rightarrow G_\delta$ such that $\frac{1}{|\binom{X}{2}|} \sum_{(u,v) \in \binom{X}{2}} |\langle g(u), g(v) \rangle - \langle u, v \rangle| \leq 3\delta r$, and the points of the grid can be represented using $L_{G_\delta} = k \log(4r/(\delta\sqrt{k}))$ bits.*

Proof. For each point $v \in X$ randomly and independently match a point $\tilde{v} = g(v)$ on the grid by rounding each of its coordinates v_i to one of the closest integral multiples of δ in such a way that $E[\tilde{v}_i] = v_i$. This distribution is given by assigning $\lceil \frac{v_i}{\delta} \rceil \delta$ with probability $p = (\frac{v_i}{\delta} - \lfloor \frac{v_i}{\delta} \rfloor)$, and assigning $\lfloor \frac{v_i}{\delta} \rfloor \delta$ with probability $1 - p$. For any $u \neq v \in X$ we have:

$$\begin{aligned}
 \mathbb{E}[|\langle \tilde{u}, \tilde{v} \rangle - \langle u, v \rangle|] & \leq \mathbb{E}[|\langle \tilde{u} - u, v \rangle|] + \mathbb{E}[|\langle \tilde{u}, \tilde{v} - v \rangle|] \\
 & \leq (\mathbb{E}[(\langle \tilde{u} - u, v \rangle)^2])^{1/2} + (\mathbb{E}[(\langle \tilde{u}, \tilde{v} - v \rangle)^2])^{1/2},
 \end{aligned}$$

where the last inequality is by Jensen's. We bound each term separately.

$$\begin{aligned}
 \mathbb{E}[(\langle \tilde{u} - u, v \rangle)^2] & = \mathbb{E} \left[\left(\sum_{i=1}^k (\tilde{u}_i - u_i) v_i \right)^2 \right] = \\
 & = \sum_{i=1}^k v_i^2 \mathbb{E}[(\tilde{u}_i - u_i)^2] + 2 \sum_{1 \leq i \neq j \leq k} v_i v_j \mathbb{E}[\tilde{u}_i - u_i] \mathbb{E}[\tilde{u}_j - u_j] \leq \delta^2 \|v\|_2^2
 \end{aligned}$$

since $|\tilde{u}_i - u_i| \leq \delta$ and $E[\tilde{u}_i] = u_i$. Similarly, for the second term we have

$$\begin{aligned}
 E [(\langle \tilde{u}, \tilde{v} - v \rangle)^2] &= E \left[\left(\sum_{i=1}^k \tilde{u}_i (\tilde{v}_i - v_i) \right)^2 \right] \leq \sum_{i=1}^k E [\tilde{u}_i^2] E [(\tilde{v}_i - v_i)^2] \\
 &+ 2 \sum_{1 \leq i \neq j \leq k} E[\tilde{u}_i \tilde{u}_j (\tilde{v}_i - v_i)] E[\tilde{v}_j - v_j] \leq \delta^2 \sum_{i=1}^k E[\tilde{u}_i^2],
 \end{aligned}
 \tag{4}$$

because the random variables \tilde{u}_i and \tilde{v}_i are independent. We also have that

$$\sum_{i=1}^k E[\tilde{u}_i^2] = \sum_{i=1}^k E[(u_i + (\tilde{u}_i - u_i))^2] = \sum_{i=1}^k (u_i^2 + 2u_i E[\tilde{u}_i - u_i] + E[(\tilde{u}_i - u_i)^2]) \leq \|u\|_2^2 + \delta^2 k.$$

Therefore, putting all together, $E[|\langle \tilde{u}, \tilde{v} \rangle - \langle u, v \rangle|] \leq \delta r + \delta(r^2 + \delta^2 k)^{1/2} \leq 2\delta r + \delta^2 \sqrt{k} \leq 3\delta r$.

The bound on the average difference in inner product in the lemma follows by the linearity of expectation, and the implied existence of a mapping with bound at most the expectation. The upper bound on the representation of the grid points was essentially given in [6]: The i th coordinate of a point x on the grid is given by a sign and an absolute value $n_i \delta$, where $0 \leq n_i \leq r/\delta$ are integers satisfying $\sum_{1 \leq i \leq k} n_i^2 \leq (r/\delta)^2$. So can be represented by the sign and their binary representation of size at most $\sum_{i=1}^k (\log(n_i) + 1)$, which is maximized when all n_i^2 's are equal, which gives the bound of $k \log(4r/(\delta\sqrt{k}))$. ◀

Combining the lemmas we obtain:

► **Corollary 9.** *For any $I \in \mathcal{P}$ let $f : I \rightarrow \ell_2^k$ be a map with $\text{Energy}_1(f) \leq \epsilon$, for $\epsilon \leq 1/36$. Let $0 < \alpha \leq 1/16$. There is $\hat{I} \subseteq I$ of size $|\hat{I}| \geq (1-\alpha)|I|$ such that there is a set $G \subset \ell_2^k$ and a map $g : \hat{I} \rightarrow G$ such that $\frac{1}{\binom{\hat{I}}{2}} \sum_{(u,v) \in \binom{\hat{I}}{2}} |\langle g(f(u)), g(f(v)) \rangle - \langle u, v \rangle| \leq (13 + \frac{0.76}{\alpha}) \epsilon$, and the points in G can be uniquely represented by binary strings of length at most $L_G = k \log(4r/(\epsilon\sqrt{k}))$ bits, where $r < 1 + 0.09 \frac{1}{\alpha}$.*

Proof. The corollary follows by applying Lemma 7 followed by Lemma 8 with $X = \hat{I}$ and $\delta = \epsilon$. Note that we may assume that $\epsilon = \delta < 1/\sqrt{k} < r/\sqrt{k}$, as otherwise we are done. ◀

We are now ready to obtain the main consequence which will imply the lower bound.

► **Corollary 10.** *For any $I \in \mathcal{P}$ let $f : I \rightarrow \ell_2^k$ be an embedding with $\text{Energy}_1(f) \leq \epsilon$, with $\epsilon \leq 1/36$. Let $0 < \alpha \leq 1/16$ and $0 < \beta$ be parameters. There is a subset of points G that satisfies the following: there is a subset $\mathcal{Y}^G \subseteq Y$ of size $|\mathcal{Y}^G| \geq (1 - 3\alpha - \frac{3}{\sqrt{2}}\beta)|Y|$ such that for each $y \in \mathcal{Y}^G$ there is a subset $\mathcal{E}_y^G \subseteq E$ of size $|\mathcal{E}_y^G| \geq (1 - 3\alpha - \frac{3}{\sqrt{2}}\beta)|E|$ such that for all pairs $(y, e) \in \mathcal{Y}^G \times \mathcal{E}_y^G$ we have: $|\langle g(f(y)), g(f(e)) \rangle - \langle y, e \rangle| \leq \frac{1}{\beta^2} (13 + \frac{0.76}{\alpha}) \epsilon$, where $g : \mathcal{Y}^G \cup \{\mathcal{E}_y^G\}_{y \in \mathcal{Y}^G} \rightarrow G$. Moreover, the points in G can be uniquely represented by binary strings of length at most $L_G = k \log(4r/(\epsilon\sqrt{k}))$ bits, where $r < 1 + 0.09 \frac{1}{\alpha}$.*

Proof. Applying Corollary 9 and Markov's inequality there are at most β^2 fraction of pairs $(u, v) \in \binom{\hat{I}}{2}$ such that $|\langle g(f(u)), g(f(v)) \rangle - \langle u, v \rangle| > \frac{1}{\beta^2} (13 + \frac{0.76}{\alpha}) \epsilon$. It follows that the number of pairs in $Y \times E$ that are in $\binom{\hat{I}}{2}$ and have this property is at most $\beta^2 \cdot \frac{3d(3d-1)}{2} \leq \frac{9}{2} \beta^2 \cdot d^2$. Therefore there can be at most $\frac{3}{\sqrt{2}} \beta d$ points in $u \in Y$ such that there are more than $\frac{3}{\sqrt{2}} \beta d$ points in $v \in E$ with this property. Since there are at most $3\alpha d$ points in each of Y and E which may not be in \hat{I} we obtain the stated bounds on the sizes of $|\mathcal{Y}^G|$ and $|\mathcal{E}_y^G|$. ◀

2.1 Encoding algorithm

Let $t = 8$. We set $\alpha = 1/(12t)$, $\beta = 1/(\sqrt{2}t)$, which implies that $r \leq 10$. Therefore, by Corollary 10, we can find a subset $G \subseteq B_2(10)$, and a mapping $g : f(I) \rightarrow G$, and a subset $\mathcal{Y}^G \subseteq Y$, with $|\mathcal{Y}^G| \geq (1 - \frac{1}{t})|Y|$, where for all $y \in \mathcal{Y}^G$ we can find a subset $\mathcal{E}_y^G \subseteq E$ with $|\mathcal{E}_y^G| \geq (1 - \frac{1}{t})|E|$, such that for all pairs $(e, y) \in \mathcal{Y}^G \times \mathcal{E}_y^G$ the inner products $|\langle g(f(y)), g(f(e)) \rangle - \langle y, e \rangle| \leq 12000\epsilon$. Moreover, each point in G can be uniquely encoded using at most $L_G = k \log(40/(\epsilon\sqrt{k}))$ bits.

We first encode all the points $Y \setminus \mathcal{Y}^G$. For each $y_S \in Y \setminus \mathcal{Y}^G$ we explicitly write down a bit for each $e_i \in E$ indicating whether $e_i \in S$. This requires d bits for each y_S and in total at most $(\frac{1}{t})d^2$ bits for the subset $Y \setminus \mathcal{Y}^G$. The next step is to encode all the points in $\{\mathcal{E}_y^G\}_{y \in \mathcal{Y}^G}$ in a way that will enable to recover all the vectors in the set together with the indices. We can do that by writing an ordered list containing d strings (one for each vector in the set E , according to its order). Each string is of length L_G bits, where each point $e_i \in \{\mathcal{E}_y^G\}_{y \in \mathcal{Y}^G}$ is encoded by its representation in G , i.e., $g(f(e_i))$, and rest of points (if there are any) are encoded by zeros. This gives an encoding of total length dL_G bits.

Now we can encode the points in \mathcal{Y}^G . Each $y_S \in \mathcal{Y}^G$ is encoded by the encoding of $g(f(y_S))$ using L_G bits, and in addition we add the encoding of the set of indices of the points in $E \setminus \mathcal{E}_{y_S}^G$, using at most $\log \binom{d}{(1/t)d} \leq (1/t)d \log(et)$ bits. Note that this information is not enough in order to recover the vector y_S , as we can't deduce whether $i \in S$ for $e_i \in E \setminus \mathcal{E}_{y_S}^G$. So we add this information explicitly, by writing down whether $i \in S$ for each $e_i \in E \setminus \mathcal{E}_{y_S}^G$, using at most $(1/t)d$ bits. In total, it takes $L_G + (1/t)d \log(et) + (1/t)d$ bits per point in \mathcal{Y}^G .

Therefore, each instance $I \in \mathcal{P}$ can be encoded using at most

$$(1/t)d^2 + dL_G + |\mathcal{Y}^G| \cdot (L_G + d(1/t) \log(et) + (1/t)d) \leq (1/t)d^2(2 + \log(et)) + 2dL_G$$

bits, since $|\mathcal{Y}^G| \leq d$. For our choice of $t = 8$, this is at most $\frac{7}{8}d^2 + 2dL_G$.

2.2 Decoding algorithm

To recover a set I given the encoding it is enough to recover the set Y , as the sets O and E are the same in each I . We first recover $Y \setminus \mathcal{Y}^G$ in a straightforward way from its naive encoding. To recover a point $y_S \in \mathcal{Y}^G$ we need to know for each $e_i \in E$ whether $i \in S$. An important implication of Corollary 10 is that given $g(f(e_i))$ and $g(f(y_S))$ of any pair $(e_i, y_S) \in \mathcal{Y}^G \times \mathcal{E}_{y_S}^G$, we can decide whether $i \in S$ by computing $\langle g(f(e_i)), g(f(y_S)) \rangle$. Recall that if $i \notin S$ then $\langle e_i, y_S \rangle = 0$, and if $i \in S$ then $\langle e_i, y_S \rangle \geq (1/2)\gamma\epsilon$. Therefore, by setting $\gamma = 48001$ we have that if $\langle g(f(e_i)), g(f(y_S)) \rangle \leq 12000\epsilon$, then $i \notin S$, and $i \in S$ otherwise. We can recover each $g(f(y_S))$ for $y_S \in \mathcal{Y}^G$ from its binary representation. Next, we recover the set of indices of the points in $E \setminus \mathcal{E}_{y_S}^G$, from which we deduce the set of indices of the points $e_i \in \mathcal{E}_{y_S}^G$. This gives the information about the set $\{g(f(e_i))\}_{e_i \in \mathcal{E}_{y_S}^G}$. At this stage we have all the necessary information to compute the inner products $\langle g(f(y_S)), g(f(e_i)) \rangle$ for all the pairs y_S and e_i that enable us to correctly decide whether $i \in S$. Finally, for the rest points $e \in E \setminus \mathcal{E}_{y_S}^G$ we have a naive encoding which explicitly states whether e is a part of y_S .

2.3 Deducing the lower bound

From the counting argument, the maximal number of different sets that can be recovered from the encoding of length at most $\rho = \frac{7}{8}d^2 + 2dL_G$ is at most 2^ρ . This implies $\frac{7}{8}d^2 + 2dL_G \geq \log|\mathcal{P}|$. On the other hand, the size of the family is $|\mathcal{P}| = \binom{d}{l}^d$. Recall that we have set $d = 2l$ so we have that $|\mathcal{P}| \geq \binom{2l}{l}^{2l} \geq \left(2^{(2l-1)}/\sqrt{l}\right)^{2l} \geq 2^{4l^2 - 2l \log l} \geq 2^{3.9l^2}$, where the last estimate

follows from our assumption on ϵ . Therefore, $\frac{7}{2}l^2 + 4LL_G \geq 3.9l^2$, implying $L_G \geq (1/10)l$, where $L_G = k \log(40/(\epsilon\sqrt{k})) = \frac{1}{2}k \log\left(16\left(\frac{10}{\epsilon}\right)^2 \frac{1}{k}\right)$. This implies that $k \log\left(16\left(\frac{10}{\epsilon}\right)^2 \frac{1}{k}\right) \geq (1/5)l \geq 1/(5\gamma^2 \cdot \epsilon^2)$. Setting $x = k \cdot (5\gamma^2 \cdot \epsilon^2)$ we have that

$$1 \leq x \log\left(\frac{0.5}{x} \cdot 2^{14}\gamma^2\right) = x \log(0.5/x) + x \log(2^{14}\gamma^2) \leq 1/2 + 2x(7 + \log \gamma),$$

where the last inequality we have used $x \log(0.5/x) \leq 0.5/(e \ln 2) < 1/2$ for all x . This implies the desired lower bound on the dimension: $k \geq 1/(20\gamma^2(7 + \log \gamma) \cdot \epsilon^2)$.

3 Lower bounds for q -moments of distortion

In this section we prove Theorem 5 which provides a lower bound for q -moments of distortion. Similarly, to the proof for ℓ_1 -distortion in Section 2, we prove the theorem first for metric space of fixed size $\hat{n} = O(1/\epsilon^2) \cdot e^{O(\epsilon q)}$, which can be extended for metric spaces of size $\Theta(n)$ for any n via metric composition [9, 8], as described in the full version of the paper.

Assume w.l.o.g that $q \geq \frac{3}{\epsilon}$, otherwise the theorem follows from Theorem 4 by monotonicity of the ℓ_q -distortion. The proof strategy has exactly the same structure as in the proof of Section 2, however the sets I are constructed using different parameters. For a given $\epsilon < 0$, let $l = \lceil \frac{1}{\gamma^2 \epsilon^2} \rceil$ be an integer for some large constant $\gamma > 1$ to be determined later. We construct point sets $I \subset \ell_2^d$, where $d = l\tau$, $\tau = e^{\epsilon q}$, and $|I| = 3d$. Assume that for all $I \in \mathcal{P}$ there is a map $f : I \rightarrow \ell_2^k$, with $\ell_q\text{-dist}(f) \leq 1 + \epsilon$. We show that this implies that $k = \Omega(q/\epsilon)$.

As before the strategy is to produce a unique binary encoding of I of length $\text{length}(I)$. We will obtain that $|\mathcal{P}| = \binom{d}{l}^d \geq (d/l)^{ld}$, which will give that $\text{length}(I) \geq dl \log(d/l) = dl \log(\tau)$. We will show that this implies the bound on $k \geq \Omega(l \log(\tau)) = \Omega(1/\epsilon^2 \cdot \epsilon q) = \Omega(q/\epsilon)$.

As in the proof of Theorem 4, we can assume w.l.o.g. that $\epsilon \leq 1/\gamma$, which by the choice of γ later on implies $\epsilon < 1/36$.

► **Lemma 11.** *For any $I \in \mathcal{P}$ let $f : I \rightarrow \ell_2^k$ be an embedding with $\ell_q\text{-dist}(f) \leq 1 + \epsilon$, for $\epsilon < 1/36$. There is a subset $\hat{I} \subseteq I$ of size $|\hat{I}| \geq (1 - 3/\tau^4)|I|$ such that $f(\hat{I}) \subset B_2(1 + 6.02\epsilon)$, and for $1 - 2/\tau^4$ fraction of the pairs $(u, v) \in \binom{\hat{I}}{2}$ it holds that $|\langle f(u), f(v) \rangle - \langle u, v \rangle| \leq 32\epsilon$.*

Proof. By assumption we have $(\ell_q\text{-dist}(f))^q = \frac{1}{\binom{I}{2}} \sum_{(u,v) \in \binom{I}{2}} (\text{dist}_f(u, v))^q \leq (1 + \epsilon)^q$.

By the Markov inequality there are at least $1 - 1/\tau^4$ fraction of the pairs $(u, v) \in \binom{I}{2}$ such that $(\text{dist}_f(u, v))^q \leq \tau^4(1 + \epsilon)^q \leq (1 + \epsilon)^q \cdot e^{4\epsilon q}$, implying that $\text{dist}_f(u, v) \leq 1 + 6\epsilon$. Therefore, $|\text{expans}_f(u, v) - 1| \leq \max\{\text{expans}_f(u, v) - 1, 1/\text{expans}_f(u, v) - 1\} = \text{dist}_f(u, v) - 1 \leq 6\epsilon$.

For every $o_j \in O$, let F_j be the set of points $v \in I \setminus \{o_j\}$ such that $|\text{expans}_f(o_j, v) - 1| > 6\epsilon$. Then the total number of pairs $(u, v) \in \binom{I}{2}$ with the property that $|\text{expans}_f(u, v) - 1| > 6\epsilon$ is at least $\sum_{j=1}^d |F_j|/2$, implying that there must be a point $\hat{o} = o_{j^*} \in O$ such that $|F_{j^*}| \leq \frac{1}{\tau^4} \cdot \frac{3d(3d-1)}{d} \leq \frac{3}{\tau^4}(3d-1)$. Define $\hat{I} = I \setminus F_{j^*}$ to be the complement of this set, so that $|\hat{I}| \leq (1 - \frac{3}{\tau^4})|I|$. We assume without loss of generality that $f(\hat{o}) = 0$. Let $\hat{O} = O \cap \hat{I}$. We have that $|\text{expans}_f(v, \hat{o}) - 1| = \frac{\|f(v)\|_2}{\|v - \hat{o}\|_2} - 1 \leq 6\epsilon$, and using $\|v - \hat{o}\|_2 \leq \|v\|_2 + \|\hat{o}\|_2 \leq 1 + \epsilon/100$, so that $\|f(v)\|_2 \leq (1 + 6\epsilon)(1 + \epsilon/100) \leq 1 + 6.02\epsilon$, implying that $f(v) \in B_2(1 + 6.02\epsilon)$.

Denote by \hat{G} the set of pairs $(u, v) \in \binom{\hat{I}}{2}$ satisfying that $|\text{expans}_f(u, v) - 1| \leq 6\epsilon$. To bound the fraction of these pairs from below, we can first bound $|\hat{I}| \geq (1 - \frac{3}{\tau^4})|I| \geq \frac{5}{2}d$ and $|\hat{I}| - 1 \geq 2d$, using that $\tau > 3$ by our assumption on q . Therefore, we have that the fraction of pairs $(u, v) \in \binom{\hat{I}}{2} \setminus \hat{G}$ is at most $\frac{1}{\tau^4} \cdot \frac{3d(3d-1)}{|\hat{I}|(|\hat{I}|-1)} \leq \frac{1}{\tau^4} \cdot \frac{9}{5} \leq \frac{2}{\tau^4}$.

13:12 Optimality of the JL Dimensionality Reduction for Practical Measures

Finally, to estimate the absolute difference in inner products over the set of pairs \hat{G} we recall some of the estimates from the proof of Section 2. For all $(u, v) \in \hat{G}$ we have:

$$\begin{aligned} |\langle f(u), f(v) \rangle - \langle u, v \rangle| &\leq \frac{1}{2} \left[\left| \|f(u)\|_2^2 - \|u\|_2^2 \right| + \left| \|f(v)\|_2^2 - \|v\|_2^2 \right| \right] \\ &\quad + \frac{1}{2} \left[\left| \|f(u) - f(v)\|_2^2 - \|u - v\|_2^2 \right| \right]. \end{aligned}$$

We can bound each term as follows:

$$\begin{aligned} \left| \|f(u)\|_2^2 - \|u\|_2^2 \right| &= \left| \|f(u) - f(\hat{\delta})\|_2^2 - \|u - \hat{\delta}\|_2^2 + \|u - \hat{\delta}\|_2^2 - \|u\|_2^2 \right| \\ &\leq \|f(u) - f(\hat{\delta})\|_2 - \|u - \hat{\delta}\|_2 \cdot (\|f(u) - f(\hat{\delta})\|_2 + \|u - \hat{\delta}\|_2) \\ &\quad + \left| \|u - \hat{\delta}\|_2 - \|u\|_2 \right| \cdot (\|u - \hat{\delta}\|_2 + \|u\|_2) \\ &\leq \|u - \hat{\delta}\|_2 |\text{expans}_f(u, \hat{\delta}) - 1| \cdot (\|f(u)\|_2 + \|u - \hat{\delta}\|_2) + \|\hat{\delta}\|_2 \cdot (\|u - \hat{\delta}\|_2 + \|u\|_2) \\ &\leq \left(1 + \frac{\epsilon}{100}\right) |\text{expans}_f(u, \hat{\delta}) - 1| \left(1 + 6.02\epsilon + 1 + \frac{\epsilon}{100}\right) + \frac{\epsilon}{100} \cdot \left(2 + \frac{\epsilon}{100}\right) \\ &\leq (2 + 6.06\epsilon) |\text{expans}_f(u, \hat{\delta}) - 1| + \frac{\epsilon}{40} \leq (2 + 6.06\epsilon) \cdot 6\epsilon + \frac{\epsilon}{40} \leq 14\epsilon, \end{aligned}$$

where we have used $\|\hat{\delta}\|_2 \leq \epsilon/100$, $\|u - \hat{\delta}\|_2 \leq \|u\|_2 + \|\hat{\delta}\|_2 \leq 1 + \epsilon/100$, the bound on the norms of the embedding within \hat{I} , and the property of pairs in \hat{G} . Additionally, we have that

$$\begin{aligned} \left| \|f(u) - f(v)\|_2^2 - \|u - v\|_2^2 \right| &= \left| \|f(u) - f(v)\|_2 - \|u - v\|_2 \right| \cdot (\|f(u) - f(v)\|_2 + \|u - v\|_2) \\ &\leq \|u - v\|_2 |\text{expans}_f(u, v) - 1| \cdot (\|f(u)\|_2 + \|f(v)\|_2 + \|u - v\|_2) \\ &\leq \sqrt{2} \left(2(1 + 6.02\epsilon) + \sqrt{2}\right) |\text{expans}_f(u, v) - 1| \leq 6|\text{expans}_f(u, v) - 1| \leq 36\epsilon, \end{aligned}$$

since $\|u - v\|_2 \leq \text{diam}(I) = \sqrt{2}$, and the last step follows using the property of pair in \hat{G} . We conclude that for all $(u, v) \in \hat{G}$: $|\langle f(u), f(v) \rangle - \langle u, v \rangle| \leq \frac{1}{2} (2 \cdot 14\epsilon + 36\epsilon) = 32\epsilon$. \blacktriangleleft

As before, the goal is to encode the images of the embedding using a sufficiently small number of bits, by rounding them to the points of a grid of the Euclidean ball via the randomized rounding technique of [6] as to preserve the inner product gap. The following lemma provides the probability that this procedure fails.

► Lemma 12. *Let $X \subset \ell_2^k$ such that $X \subset B_2(r)$. For $\delta \leq r/\sqrt{k}$ let $G_\delta \subseteq B_2(r)$ denote the intersection of the δ -grid with $B_2(r)$. There is a mapping $g : X \rightarrow G_\delta$ such that for any $\eta \geq 1$, there is a $1 - 4e^{-\eta^2}$ fraction of the pairs $(u, v) \in \binom{X}{2}$ such that $|\langle g(u), g(v) \rangle - \langle u, v \rangle| \leq 3\sqrt{2}\eta\delta r$, and the points of the grid can be represented using $L_{G_\delta} = k \log(4r/(\delta\sqrt{k}))$ bits.*

Proof. For each point $v \in X$ randomly and independently match a point \tilde{v} on the grid by rounding each of its coordinates v_i to one of the closest integral multiples of δ in such a way that $E[\tilde{v}_i] = v_i$. For any $u \neq v \in X$ we have: $|\langle \tilde{u}, \tilde{v} \rangle - \langle u, v \rangle| \leq |\langle \tilde{u} - u, v \rangle| + |\langle \tilde{u}, \tilde{v} - v \rangle|$. Now, $E[\langle \tilde{u} - u, v \rangle] = \sum_{i=1}^k E[\tilde{u}_i - u_i]v_i = 0$ and $E[\langle \tilde{u}, \tilde{v} - v \rangle] = \sum_{i=1}^k E[\tilde{u}_i]E[\tilde{v}_i - v_i] = 0$. Next, we wish to make use of the Hoeffding bound. We therefore bound each of the terms $|\langle \tilde{u} - u, v \rangle| \leq \delta \|v\|_2$ and the sum $\sum_{i=1}^k \delta^2 v_i^2 = \delta^2 r$, and $|\tilde{u}_i(\tilde{v}_i - v_i)| \leq \delta(|u_i| + \delta)$, so that

$$\sum_{i=1}^k \delta^2 (v_i + \delta)^2 = \delta^2 \sum_{i=1}^k (v_i^2 + 2\delta v_i + \delta^2) \leq \delta^2 (r + 2\delta \|v\|_1 + \delta^2 k) \leq \delta^2 (r^2 + 2\delta r\sqrt{k} + \delta^2 k) \leq 4\delta^2 r^2.$$

Applying the Hoeffding bound we get that $\Pr[|\langle \tilde{u} - u, v \rangle| > \sqrt{2}\eta\delta r] \leq 2e^{-\eta^2}$ and $\Pr[|\langle \tilde{u}, \tilde{v} - v \rangle| > 2\sqrt{2}\eta\delta r] \leq 2e^{-\eta^2}$, and therefore $\Pr[|\langle \tilde{u}, \tilde{v} \rangle - \langle u, v \rangle| > 3\sqrt{2}\eta\delta r] \leq 4e^{-\eta^2}$. This

probability also bounds the expected number of pairs with this property so there must exist an embedding to the grid where the bound stated in the lemma holds. The bound on the representation size is the same as in Lemma 8. ◀

Combining the lemmas we obtain:

► **Corollary 13.** *For any $I \in \mathcal{P}$ let $f : I \rightarrow \ell_2^k$ be an embedding with $\ell_q\text{-dist}(f) \leq 1 + \epsilon$, with $\epsilon \leq 1/36$. There is a subset $\hat{I} \subseteq I$ of size $|\hat{I}| \geq (1 - 3/\tau^4)|I|$ such that for a fraction of at least $1 - 6/\tau^4$ of the pairs $(u, v) \in \binom{\hat{I}}{2}$ it holds that: $|\langle g(f(u)), g(f(v)) \rangle - \langle u, v \rangle| \leq 42\epsilon$, where $g : \hat{I} \rightarrow G$. Moreover, the points in G can be uniquely represented by binary strings of length at most $L_G = k \log(5\sqrt{q}/(\epsilon k))$ bits.*

Proof. The corollary follows by applying Lemma 11 followed by Lemma 12 with $X = \hat{I}$ with $\delta = 2\sqrt{\epsilon/q}$ and $\eta = \sqrt{\ln(\tau)}$. Note that we may assume that indeed $2\sqrt{\epsilon/q} = \delta < 1/\sqrt{k} < r/\sqrt{k}$, since otherwise we are done. Therefore, the increase in the absolute difference of the inner products due to the grid embedding is at most: $3\sqrt{2}\eta\delta r = 6r\sqrt{2\ln(\tau)\epsilon/q} = 6r\sqrt{2(\epsilon q)\epsilon/q} \leq 10\epsilon$. The bound on representation of the grid follows from Lemma 12: $L_G = k \log(4r/(\delta\sqrt{k})) = k \log(4r\sqrt{q}/(\epsilon k)) \leq k \log(5\sqrt{q}/(\epsilon k))$. ◀

We are ready to obtain the main technical consequence which will imply the lower bound:

► **Corollary 14.** *For any $I \in \mathcal{P}$ let $f : I \rightarrow \ell_2^k$ be an embedding with $\ell_q\text{-dist}(f) \leq \epsilon$, with $\epsilon \leq 1/36$. There is a subset of points G that satisfies the following: there is a subset $\mathcal{Y}^G \subseteq Y$ of size $|\mathcal{Y}^G| \geq (1 - 6/\tau^2)|Y|$ such that for each $y \in \mathcal{Y}^G$ there is a subset $\mathcal{E}_y^G \subseteq E$ of size $|\mathcal{E}_y^G| \geq (1 - 6/\tau^2)|E|$ such that for all pairs $(y, e) \in \mathcal{Y}^G \times \mathcal{E}_y^G$ we have: $|\langle g(f(y)), g(f(e)) \rangle - \langle y, e \rangle| \leq 42\epsilon$, where $g : \mathcal{Y}^G \cup \{\mathcal{E}_y^G\}_{y \in \mathcal{Y}^G} \rightarrow G$. Moreover, the points in G can be uniquely represented by binary strings of length at most $L_G = k \log(5\sqrt{q}/(\epsilon k))$ bits.*

Proof. Applying Corollary 13 we have that there are at most $6/\tau^4$ pairs $(u, v) \in \binom{\hat{I}}{2}$ such that $|\langle g(f(u)), g(f(v)) \rangle - \langle u, v \rangle| > 42\epsilon$. It follows that the number of pairs in $Y \times E$ that are in $\binom{\hat{I}}{2}$ and have this property is at most $\frac{6}{\tau^4} \cdot \frac{3d(3d-1)}{2} \leq \frac{27}{\tau^4} \cdot d^2$. Therefore there can be at most $\frac{3\sqrt{3}}{\tau^2} \cdot d$ points in $u \in Y$ such that there are more than $\frac{3\sqrt{3}}{\tau^2}d$ points in $v \in E$ with this property. Since there at most $\frac{3}{\tau^4} \cdot d < \frac{0.5}{\tau^2} \cdot d$ points in each Y and E which may not be in \hat{I} we obtain the stated bounds on the sizes of $|\mathcal{Y}^G|$ and $|\mathcal{E}_y^G|$. ◀

3.1 Encoding and decoding

For a set $I \in \mathcal{P}$ let $f : I \rightarrow \ell_2^k$ be a map with $\ell_q\text{-dist}(f) = 1 + \epsilon$, where $\Omega\left(\frac{1}{\sqrt{n}}\right) \leq \epsilon < 1/36$, and $q = O(\log(\epsilon^2 n)/\epsilon)$. Let $t = \tau^2/6$. Therefore, by Corollary 14, we can find a subset $G \subseteq B_2(2)$, and a mapping $g : f(I) \rightarrow G$, and a subset $\mathcal{Y}^G \subseteq Y$, with $|\mathcal{Y}^G| \geq (1 - \frac{1}{t})|Y|$, where for all $y \in \mathcal{Y}^G$ we can find a subset $\mathcal{E}_y^G \subseteq E$ with $|\mathcal{E}_y^G| \geq (1 - \frac{1}{t})|E|$, such that for all pairs $(y, e) \in \mathcal{Y}^G \times \mathcal{E}_y^G$ the inner products $|\langle g(f(y)), g(f(e)) \rangle - \langle y, e \rangle| \leq 42\epsilon$. Moreover, each point in G can be uniquely encoded using at most $L_G = k \log(5\sqrt{q}/(\epsilon k))$ bits.

The encoding is done according to the description in Section 2.1 so we similarly obtain the following bound on the bit length of the encoding: $(1/t)d^2(2 + \log(et)) + 2dL_G$.

The decoding works in the same way as before for an appropriate choice of $\gamma = 169$.

3.2 Deducing the lower bound

In this subsection we show that $k = \Omega(q/\epsilon)$, proving the desired lower bound for the case of $n = 3d = O(1/\epsilon^2) \cdot e^{O(\epsilon q)}$. From the counting argument, the maximal number of different sets that can be recovered from the encoding of length at most $\rho = (1/t)d^2(2 + \log(et)) + 2dL_G$ is at most 2^ρ . This implies $(1/t)d^2(2 + \log(et)) + 2dL_G \geq \log|\mathcal{P}|$. On the other hand, the size of the family is $|\mathcal{P}| = \binom{d}{t}^d \geq (d/t)^{ld} = \tau^{ld}$, so that $\log(|\mathcal{P}|) = ld \log(\tau)$. We therefore derive the following inequality

$$(1/t)d^2(2 + \log(et)) + 2dL_G \geq ld \log(\tau) \Rightarrow L_G \geq (1/4)l \log(\tau),$$

as $(1/t)d(2 + \log(et)) \leq d(2 \log(\tau) + 4)/\tau^2 \leq d/(2\tau) \log(\tau) = l \log(\tau)/2$, using that $\log(\tau) > 4$.

Recall that $L_G = k \log(5\sqrt{q}/(\epsilon k)) = \frac{1}{2}k \log(25(q/(\epsilon k)))$. This implies that

$$k \log\left(25 \left(\frac{q}{\epsilon k}\right)\right) \geq (1/2)l \log(\tau) \geq 1/(2\gamma^2 \cdot \epsilon^2) \cdot \epsilon q = 1/(2\gamma^2) \cdot q/\epsilon.$$

Setting $x = k \cdot (2\gamma^2 \cdot \epsilon/q)$ we have that

$$1 \leq x \log\left(\frac{0.5}{x} \cdot 100\gamma^2\right) = x \log(0.5/x) + x \log(100\gamma^2) \leq 1/2 + 2x \log(10\gamma),$$

where the last inequality we have used $x \log(0.5/x) \leq 0.5/(e \ln 2) < 1/2$ for all x . This implies the desired lower bound on the dimension: $k \geq 1/(8\gamma^2 \log(10\gamma)) \cdot q/\epsilon$.

References

- 1 Ittai Abraham, Yair Bartal, T-H. Hubert Chan, Kedar Dhamdhere Dhamdhere, Anupam Gupta, Jon Kleinberg, Ofer Neiman, and Aleksandrs Slivkins. Metric embeddings with relaxed guarantees. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS '05*, pages 83–100, USA, 2005. IEEE Computer Society. doi:10.1109/SFCS.2005.51.
- 2 Ittai Abraham, Yair Bartal, and Ofer Neiman. Embedding metrics into ultrametrics and graphs into spanning trees with constant average distortion. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07*, pages 502–511, USA, 2007. Society for Industrial and Applied Mathematics.
- 3 Ittai Abraham, Yair Bartal, and Ofer Neiman. Embedding metrics into ultrametrics and graphs into spanning trees with constant average distortion. In *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms, SODA '07*, pages 502–511, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. URL: <http://portal.acm.org/citation.cfm?id=1283383.1283437>.
- 4 Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026–3126, 2011. doi:10.1016/j.aim.2011.08.003.
- 5 Noga Alon. Perturbed identity matrices have high rank: Proof and applications. *Combinatorics, Probability & Computing*, 18(1-2):3–15, 2009. doi:10.1017/S0963548307008917.
- 6 Noga Alon and Bo'az Klartag. Optimal compression of approximate inner products and dimension reduction. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–650, 2017. doi:10.1109/FOCS.2017.65.
- 7 Ehsan Amid and Manfred K. Warmuth. Trimap: Large-scale dimensionality reduction using triplets, 2019. arXiv:1910.00204.
- 8 Yair Bartal, Nova Fandina, and Ofer Neiman. Dimensionality reduction: theoretical perspective on practical measures. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10576–10588, 2019. URL: <https://proceedings.neurips.cc/paper/2019/hash/94f4ede62112b790c91d5e64fdb09cb8-Abstract.html>.

- 9 Yair Bartal, Nathan Linial, Manor Mendel, and Assaf Naor. On metric ramsey-type phenomena. *Annals of Mathematics*, 162(2):643–709, 2005. URL: <http://www.jstor.org/stable/20159927>.
- 10 A. Censi and D. Scaramuzza. Calibration by correlation using metric embedding from nonmetric similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2357–2370, October 2013. doi:10.1109/TPAMI.2013.34.
- 11 Leena Chennuru Vankadara and Ulrike von Luxburg. Measures of distortion for machine learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4891–4900. Curran Associates, Inc., 2018.
- 12 Leena Chennuru Vankadara and Ulrike von Luxburg. Measures of distortion for machine learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL: <https://proceedings.neurips.cc/paper/2018/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf>.
- 13 Russ Cox, Frank Dabek, Frans Kaashoek, Jinyang Li, and Robert Morris. Practical, distributed network coordinates. *SIGCOMM Comput. Commun. Rev.*, 34(1):113–118, January 2004. doi:10.1145/972374.972394.
- 14 Michael Elkin, Arnold Filtser, and Ofer Neiman. Prioritized metric structures and embedding. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 489–498, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2746539.2746623.
- 15 Michael Elkin, Arnold Filtser, and Ofer Neiman. Terminal Embeddings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 242–264, 2015.
- 16 Patrick J. F. Groenen, Rudolf Mathar, and Willem J. Heiser. The majorization approach to multidimensional scaling for minkowski distances. *Journal of Classification*, 12(1):3–19, 1995.
- 17 W. J Heiser. Multidimensional scaling with least absolute residuals. In *In H. H. Bock (Ed.) Classification and related methods*, pages 455–462. Amsterdam: NorthHolland, 1988a.
- 18 P. Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, page 10, USA, 2001. IEEE Computer Society.
- 19 Piotr Indyk and Jiri Matousek. Low-distortion embeddings of finite metric spaces. In *in Handbook of Discrete and Computational Geometry*, pages 177–196. CRC Press, 2004.
- 20 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM. doi:10.1145/276698.276876.
- 21 William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, pages 189–206. American Mathematical Society, Providence, RI, 1984.
- 22 Jon Kleinberg, Aleksandrs Slivkins, and Tom Wexler. Triangulation and embedding using small sets of beacons. *J. ACM*, 56(6):32:1–32:37, September 2009. doi:10.1145/1568318.1568322.
- 23 Deepanshu Kush, Aleksandar Nikolov, and Haohua Tang. Near neighbor search via efficient average distortion embeddings. In *37th International Symposium on Computational Geometry, SoCG 2021, June 7-11, 2021, Buffalo, NY, USA (Virtual Conference)*, pages 50:1–50:14, 2021. doi:10.4230/LIPIcs.SocG.2021.50.
- 24 Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 633–638, 2017. doi:10.1109/FOCS.2017.64.

13:16 Optimality of the JL Dimensionality Reduction for Practical Measures

- 25 N. Linial. Finite metric spaces- combinatorics, geometry and algorithms. In *Proceedings of the ICM*, 2002.
- 26 Jiří Matoušek. Bi-Lipschitz embeddings into low-dimensional Euclidean spaces. *Commentat. Math. Univ. Carol.*, 31(3):589–600, 1990.
- 27 Jiří Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.
- 28 Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. doi:10.21105/joss.00861.
- 29 Assaf Naor. Comparison of metric spectral gaps. *Analysis and Geometry in Metric Spaces*, 2:2:1–52, 2014.
- 30 Assaf Naor. An average John theorem. *Geometry and Topology*, 25(4):1631–1717, 2021. doi:10.2140/gt.2021.25.1631.
- 31 Puneet Sharma, Zhichen Xu, Sujata Banerjee, and Sung-Ju Lee. Estimating network proximity and latency. *Computer Communication Review*, 36(3):39–50, 2006. doi:10.1145/1140086.1140092.
- 32 Yuval Shavitt and Tomer Tanel. Big-bang simulation for embedding network distances in euclidean space. *IEEE/ACM Trans. Netw.*, 12(6):993–1006, December 2004. doi:10.1109/TNET.2004.838597.
- 33 Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL: <http://jmlr.org/papers/v9/vandermaaten08a.html>.
- 34 Santosh Srinivas Vempala. *The random projection method*, volume 65 of *DIMACS series in discrete mathematics and theoretical computer science*. Providence, R.I. American Mathematical Society, 2004. URL: <http://opac.inria.fr/record=b11101689>.
- 35 J. Fernando Vera, Willem J. Heiser, and Alex Murillo. Global optimization in any minkowski metric: A permutation-translation simulated annealing algorithm for multidimensional scaling. *J. Classif.*, 24(2):277–301, September 2007.
- 36 Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization, 2020. arXiv:2012.04456.

Quasi-Universality of Reeb Graph Distances

Ulrich Bauer  

Department of Mathematics and Munich Data Science Institute,
Technische Universität München, Germany

Håvard Bakke Bjerkevik  

Institute of Geometry, Technische Universität Graz, Austria

Benedikt Fluhr  

Department of Mathematics, Technische Universität München, Germany

Abstract

We establish bi-Lipschitz bounds certifying quasi-universality (universality up to a constant factor) for various distances between Reeb graphs: the interleaving distance, the functional distortion distance, and the functional contortion distance. The definition of the latter distance is a novel contribution, and for the special case of contour trees we also prove strict universality of this distance. Furthermore, we prove that for the special case of merge trees the functional contortion distance coincides with the interleaving distance, yielding universality of all four distances in this case.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Mathematics of computing → Trees; Theory of computation → Computational geometry

Keywords and phrases Reeb graphs, contour trees, merge trees, distances, universality, interleaving distance, functional distortion distance, functional contortion distance

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.14

Related Version *Full Version:* <https://arxiv.org/abs/2112.00720> [1]

Funding *Ulrich Bauer:* DFG SFB/TRR 109 *Discretization in Geometry and Dynamics.*

Håvard Bakke Bjerkevik: Austrian Science Fund (FWF) grant number P 33765-N.

1 Introduction

The *Reeb graph* is a topological signature of real-valued functions, first considered in the context of Morse theory [11] and subsequently applied to the analysis of geometric shapes [9, 13]. It describes the connected components of level sets of a function, and for Morse functions on compact manifolds or PL functions on compact polyhedra it turns out to be a finite topological graph with a function that is monotonic on the edges. If the domain of the function is simply-connected, then the Reeb graph is contractible, hence a tree, and is therefore often called a *contour tree*. In topological data analysis, Reeb graphs are used for surveying functions, and also in a discretized form termed *Mapper* [14] for the analysis of point cloud data, typically high-dimensional or given as an abstract finite metric space.

An important requirement for topological signatures is the ability to quantify their similarity, which is typically achieved by means of an extended pseudometric on the set of isomorphism classes of signatures under consideration, referred to as a distance. In order for such a distance to be practical, it should be resilient to noise and perturbations of the input data, which is formalized by the property of *stability*: small perturbations of the data lead to small perturbations of the signature. Mathematically speaking, the signature is a Lipschitz-continuous map between metric spaces, and often the Lipschitz constant is assumed to be 1, meaning that the map is non-expansive. Previous examples of distances between Reeb graphs satisfying stability include the *functional distortion distance* [2], the *interleaving distance* [8], and the *Reeb graph edit distance* [4]. While stability guarantees that similarity



© Ulrich Bauer, Håvard Bakke Bjerkevik, and Benedikt Fluhr;
licensed under Creative Commons License CC-BY 4.0

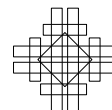
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 14; pp. 14:1–14:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



of data sets is preserved, it does not provide any guarantees regarding the *discriminativity* of the distance on the signature. Indeed, a certain loss of information is inherent and even desired for most signatures; in fact, a key strength of topological signatures is their invariance to reparametrizations or isometries of the input data, independent of the metric used to distinguish non-isomorphic signatures. Thus, given any signature map defined on some metric space of possible data, such as the space of real-valued functions on a fixed domain with the uniform metric, one stable distance is considered more discriminative than another if it assigns larger or equal distances to all possible pairs of signatures. For example, the functional distortion distance is an upper bound for the interleaving distance and thus more discriminative in that sense; in fact, the two distances are bi-Lipschitz equivalent [5]. One may now ask if a given distance is *universal*, meaning that it is both stable and an upper bound for all stable distances, and thus the most discriminative among all stable distances. This can be expressed by the universal property of a quotient metric [7, 12], giving rise to the name “universal”. Since there is only one such distance, we refer to it as *the universal distance*. Perhaps surprisingly, neither the interleaving distance nor the functional distortion distance are universal, while the *Reeb graph edit distance* [4] turns out to be universal.

These results raise the question of whether the mentioned distances are *quasi-universal*, i.e., bi-Lipschitz equivalent to the universal distance. We address this question by proving lower and upper Lipschitz bounds relating all three mentioned distances, together with the novel *functional contortion distance*, a slight variation of the functional distortion distance. It has a simple definition, is more discriminative than the functional distortion and interleaving distances while still being stable, and in fact coincides with the universal distance when restricted to contour trees, as we also show in this paper. Furthermore, we show that the interleaving distance of merge trees, considered as a special case of Reeb graphs, coincides with the functional contortion distance, establishing the universality of all four distances in this particular setting.

Previous results relating the distances considered in this paper were obtained in [2, 3, 5, 8]. We discuss these results in detail in Section 2. In [4], an edit distance is introduced and shown to be universal, and an example is given showing that the functional distortion distance is not universal. Furthermore, in [6], an ℓ^p -generalization of the interleaving distance is introduced for unbounded merge trees with finitely many nodes, for all $p \in [1, \infty]$. It satisfies a universal property analogous to the one considered. The case $p = \infty$ yields a variant of our universality result for the interleaving distance between unbounded merge trees with finitely many nodes.

2 Preliminaries

► **Definition 1** (Reeb graph). *A Reeb graph is a pair (F, f) where F is a non-empty connected topological space and $f: F \rightarrow \mathbb{R}$ a continuous function, such that F admits the structure of a 1-dimensional CW complex for which*

- *f restricts to an embedding on each 1-cell, and*
- *for every bounded interval $I \subset \mathbb{R}$, the preimage $f^{-1}(I)$ intersects a finite number of cells.*

We often refer to F as a Reeb graph without referring explicitly to the function f . A morphism (isomorphism) of Reeb graphs is a value-preserving continuous map (homeomorphism).

► **Remark 2.** Suppose (F, f) is a Reeb graph, let $I \subseteq \mathbb{R}$ be a closed interval, and fix the structure of a CW complex on F as in Definition 1. As we may subdivide any 1-cell whose interior intersects $f^{-1}(\partial I)$, the preimage $f^{-1}(I)$ also admits the structure of a CW complex. Thus, the preimage $f^{-1}(I)$ is locally path-connected and therefore the connected components and the path-components of $f^{-1}(I)$ coincide.

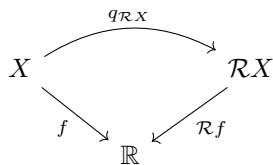
► **Definition 3** (Contour tree). *A contour tree is a contractible Reeb graph.*

Contour trees further specialize to *merge trees*, which can be thought of as upside down trees in the sense that its root is at the top and its branches grow from top to bottom.

► **Definition 4** (Merge tree). *A merge tree is a Reeb graph (F, f) , such that F admits the structure of a 1-dimensional CW complex as in Definition 1 with the additional property that each 0-cell is the lower boundary point of at most a single 1-cell.*

We note that this definition allows for both unbounded merge trees, which is an implied necessity of the definition in [10], and bounded merge trees.

► **Definition 5** (Induced Reeb graph). *Let X be a topological space, $f: X \rightarrow \mathbb{R}$ a continuous function. Let $\mathcal{R}X$ be the quotient space X/\sim_f , with $x \sim_f y$ iff x and y belong to the same connected component of some level set of $f: X \rightarrow \mathbb{R}$, and let $q_{\mathcal{R}X}: X \rightarrow \mathcal{R}X$ be the natural quotient map, and let $\mathcal{R}f: \mathcal{R}X \rightarrow \mathbb{R}$ be the unique continuous map such that the diagram*



commutes. If $(\mathcal{R}X, \mathcal{R}f)$ is a Reeb graph, we say that it is the Reeb graph induced by (X, f) .

We say that two or more points are *connected* in some space if there is a connected component of that space containing all those points. For $t \in \mathbb{R}$ and $\delta \geq 0$, let

$$I_\delta(t) := [t - \delta, t + \delta] \subset \mathbb{R}$$

denote the closed interval of radius δ centered at t .

Let $f: X \rightarrow \mathbb{R}$ be a continuous function and let $\delta \geq 0$. We define the δ -thickening of X as

$$\mathcal{T}_\delta X := X \times [-\delta, \delta], \quad \mathcal{T}_\delta f: \mathcal{T}_\delta X \rightarrow \mathbb{R}, (p, t) \mapsto f(p) + t.$$

Moreover, let

$$\tau_X^\delta: X \rightarrow \mathcal{T}_\delta X, p \mapsto (p, 0)$$

be the natural embedding of X into its δ -thickening. Now let $g: G \rightarrow \mathbb{R}$ be a Reeb graph and consider its δ -thickening $\mathcal{T}_\delta g: \mathcal{T}_\delta G \rightarrow \mathbb{R}$. We define the δ -smoothing of G as

$$\mathcal{U}_\delta G := \mathcal{R}\mathcal{T}_\delta G, \quad \mathcal{U}_\delta g := \mathcal{R}\mathcal{T}_\delta g: \mathcal{R}\mathcal{T}_\delta G \rightarrow \mathbb{R}.$$

Moreover, let

$$q_{\mathcal{U}_\delta G}: \mathcal{T}_\delta G \rightarrow \mathcal{U}_\delta G = \mathcal{R}\mathcal{T}_\delta G$$

be the natural quotient map as in Definition 5, and let

$$\zeta_G^\delta: G \xrightarrow{\tau_G^\delta} \mathcal{T}_\delta G \xrightarrow{q_{\mathcal{U}_\delta G}} \mathcal{U}_\delta G$$

be the natural map of G into its δ -smoothing, which is the composition of τ_G^δ and $q_{\mathcal{U}_\delta G}$.

14:4 Quasi-Universality of Reeb Graph Distances

Now suppose $f: F \rightarrow \mathbb{R}$ is another Reeb graph and that $\phi: F \rightarrow \mathcal{U}_\delta G$ is a continuous map. Identifying points in $\mathcal{U}_\delta G = \mathcal{RT}_\delta G$ with subsets of $\mathcal{T}_\delta G$ via the quotient map $q_{\mathcal{R}G}: \mathcal{T}_\delta G \rightarrow \mathcal{U}_\delta G$, the map ϕ induces a set-valued map

$$\Phi := \text{pr}_G \circ \phi: F \rightarrow \mathcal{P}(G)$$

from F to the power set of G , where $\text{pr}_G: \mathcal{T}_\delta G = G \times [-\delta, \delta] \rightarrow G$, $(q, t) \mapsto q$ is the projection onto G . Moreover, suppose $\psi: G \rightarrow \mathcal{U}_\delta F$ is another continuous map, and define the set-valued map analogously as

$$\Psi := \text{pr}_F \circ \psi: G \rightarrow \mathcal{P}(F).$$

► **Definition 6** (Interleaving distance d_I [8]). *We say that the pair of maps $\phi: F \rightarrow \mathcal{U}_\delta G$ and $\psi: G \rightarrow \mathcal{U}_\delta F$ is a δ -interleaving of (F, f) and (G, g) if the triangles*

$$\begin{array}{ccc} F & \xrightarrow{\phi} & \mathcal{U}_\delta G \\ & \searrow f & \swarrow \mathcal{U}_\delta g \\ & & \mathbb{R} \end{array} \quad \text{and} \quad \begin{array}{ccc} G & \xrightarrow{\psi} & \mathcal{U}_\delta F \\ & \searrow g & \swarrow \mathcal{U}_\delta f \\ & & \mathbb{R} \end{array}$$

commute and the following two conditions are satisfied:

- For any $x \in F$, x and $\Psi(\Phi(x))$ are connected in $f^{-1}(I_{2\delta}(f(x)))$.
- For any $y \in G$, y and $\Phi(\Psi(y))$ are connected in $g^{-1}(I_{2\delta}(g(y)))$.

The interleaving distance, denoted $d_I(F, G)$, is defined as the infimum of the set of δ admitting a δ -interleaving between (F, f) and (G, g) .

Note that for any $t \in \mathbb{R}$ the map $f^{-1}(I_\delta(t)) \rightarrow (\mathcal{T}_\delta f)^{-1}(t)$ given by $x \mapsto (x, t - f(x))$ is a homeomorphism, with the inverse given by the restriction of $\text{pr}_F: \mathcal{T}_\delta F \rightarrow F$. In particular, the points of $\mathcal{U}_\delta F$, which are the connected components of level sets of $\mathcal{T}_\delta f$, are in bijection with connected components of interlevel sets of f . Hence, the connectedness condition for an interleaving is equivalent to requiring that $\psi_\delta \circ \phi = \tau_F^{2\delta}$ and $\phi_\delta \circ \psi = \tau_G^{2\delta}$, where ϕ_δ is the induced map $\mathcal{U}_\delta F \rightarrow \mathcal{U}_{2\delta} G$, $[x, s] \mapsto [\phi(x), s]$ and similarly for ψ_δ .

► **Definition 7** (Functional distortion distance d_{FD} [2]). *Let (F, f) and (G, g) be two Reeb graphs. Given a pair (ϕ, ψ) of maps $\phi: F \rightarrow G$ and $\psi: G \rightarrow F$, consider the correspondence*

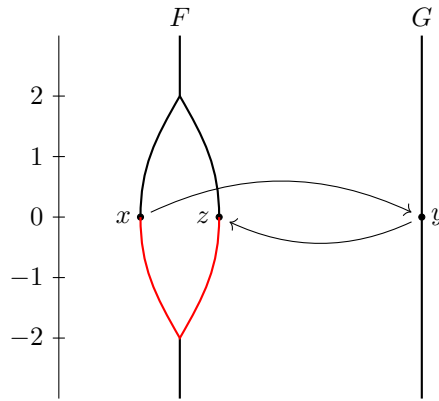
$$C(\phi, \psi) = \{(x, y) \in F \times G \mid \phi(x) = y \text{ or } x = \psi(y)\}.$$

The pair (ϕ, ψ) is a δ -distortion pair if $\|f - g \circ \phi\|_\infty \leq \delta$, $\|f \circ \psi - g\|_\infty \leq \delta$, and for all $(x, y), (x', y') \in C(\phi, \psi)$ we have

$$|d_f(x, x') - d_g(y, y')| \leq 2\delta,$$

where $d_f(x, x')$ denotes the infimum length of any interval I such that x and x' are connected in $f^{-1}(I)$, and similarly for d_g . The functional distortion distance, denoted $d_{FD}(F, G)$, is defined as the infimum of all δ admitting a δ -distortion pair between (F, f) and (G, g) .

► **Definition 8** (Functional contortion distance d_{FC}). *Let (F, f) and (G, g) be two Reeb graphs. A pair (ϕ, ψ) of maps $\phi: F \rightarrow G$ and $\psi: G \rightarrow F$ is a δ -contortion pair between (F, f) and (G, g) if the following symmetric conditions are satisfied.*



■ **Figure 1** Reeb graphs (F, f) and (G, g) . If $\phi: F \rightarrow G$ and $\psi: G \rightarrow F$ are a 1-distortion pair, we allow $\phi(x) = y$ and $\psi(y) = z$ because there is a path from x to z in $f^{-1}[-2, 0]$. However, in this case (ϕ, ψ) is not a 1-contortion pair, because x and z are not connected in $f^{-1}(I_1(g(y))) = f^{-1}[-1, 1]$.

- For any $x \in F$ and $y \in \psi^{-1}(x)$, the points $\phi(x)$ and y are connected in $g^{-1}(I_\delta(f(x)))$.
- For any $y \in G$ and $x \in \phi^{-1}(y)$, the points $\psi(y)$ and x are connected in $f^{-1}(I_\delta(g(y)))$.

The functional contortion distance, denoted $d_{FC}(F, G)$, is defined as the infimum of the set of δ admitting a δ -contortion pair between (F, f) and (G, g) .

The definition of d_{FC} is arguably easier to work with than that of d_{FD} , since to verify that (ϕ, ψ) is a δ -contortion pair, one only has to check one condition for each element of $C(\phi, \psi)$, while to verify that (ϕ, ψ) is a δ -distortion pair, one needs to check a condition for each pair of elements of $C(\phi, \psi)$. We prove that d_{FC} satisfies the triangle inequality in [1, Appendix A]. In [1, Appendix B], we give a simple example showing that d_{FC} and d_{FD} are not the same.

► **Remark 9.** Let (ϕ, ψ) be a δ -contortion pair between (F, f) and (G, g) . For any $x \in F$ we have $\phi(x) \in g^{-1}(I_\delta(f(x)))$, which implies $\|f(x) - g \circ \phi(x)\| \leq \delta$. It follows that $\|f - g \circ \phi\|_\infty \leq \delta$, and by a symmetric argument we also get $\|g - f \circ \psi\|_\infty \leq \delta$.

► **Definition 10** (Universal distance d_U [7, 4]). Let (F, f) and (G, g) be two Reeb graphs. The universal distance, denoted $d_U(F, G)$, is defined as the infimum of $\|\tilde{f} - \tilde{g}\|_\infty$ taken over all spaces Z and functions $\tilde{f}, \tilde{g}: Z \rightarrow \mathbb{R}$ such that $(\mathcal{R}Z, \mathcal{R}\tilde{f}) \cong (F, f)$ and $(\mathcal{R}Z, \mathcal{R}\tilde{g}) \cong (G, g)$.

The distance d_U is readily seen to be universal. Recall that the Reeb graph edit distance [4] is also universal, providing an alternative explicit construction for the universal distance.

If d and d' are distances on Reeb graphs and $c \in [0, \infty)$, we use the notation $d \leq cd'$ to express that for all Reeb graphs (F, f) and (G, g) , the inequality $d(F, G) \leq cd'(F, G)$ holds.

► **Theorem 11** (Quasi-universality of Reeb graph distances). The functional contortion distance d_{FC} , the functional distortion distance d_{FD} , and the interleaving distance d_I on Reeb graphs are quasi-universal (bi-Lipschitz equivalent to the universal distance). Specifically, we have

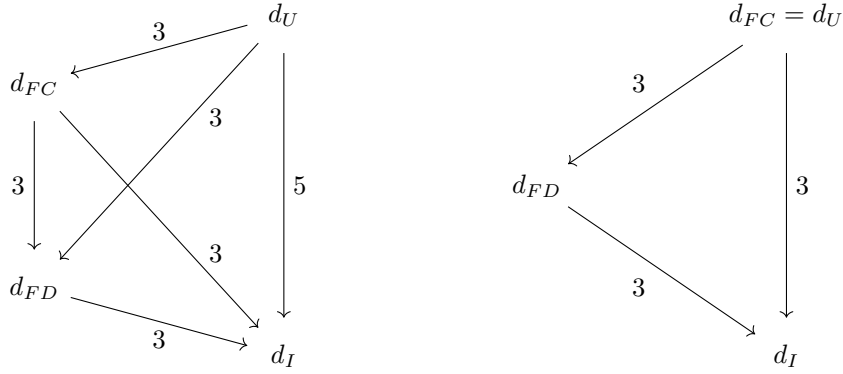
$$\begin{aligned} d_{FC} &\leq d_U \leq 3d_{FC} & d_{FD} &\leq d_U \leq 3d_{FD} & d_I &\leq d_U \leq 5d_I \\ d_{FD} &\leq d_{FC} \leq 3d_{FD} & d_I &\leq d_{FC} \leq 3d_I & d_I &\leq d_{FD} \leq 3d_I. \end{aligned}$$

Of these, only $d_I \leq d_{FD} \leq 3d_I$ [5], $d_I \leq d_U$, and $d_{FD} \leq d_U$ were known, the latter two being equivalent to stability of d_I [8] and of d_{FD} [2]. See Figure 2a for a visualization of the inequalities in Theorem 11. In fact, one can easily check that all the inequalities in the theorem follow from only six of them, namely

$$d_I \leq d_{FD} \leq d_{FC} \leq d_U \quad d_U \leq 3d_{FD} \quad d_U \leq 5d_I \quad d_{FC} \leq 3d_I.$$

In future work, we plan to show that all these bounds are indeed tight.

14:6 Quasi-Universality of Reeb Graph Distances



(a) Inequalities for Reeb graphs.

(b) Inequalities for contour trees.

■ **Figure 2** An arrow from d to d' labeled c means that the inequalities $d'(F, G) \leq d(F, G) \leq cd'(F, G)$ hold for all Reeb graphs (in (a)) or contour trees (in (b)).

► **Theorem 12** (Universality of the functional contortion distance for contour trees). *The functional contortion distance is universal for contour trees: given two contour trees (F, f) and (G, g) , we have*

$$d_{FC}(F, G) = d_U(F, G).$$

This theorem gives us a simpler set of inequalities for contour trees than what we have for general Reeb graphs; see Figure 2b. Finally, for merge trees, the hierarchy collapses:

► **Theorem 13** (Universality of the interleaving distance for merge trees). *The interleaving distance is universal for merge trees: given two merge trees (F, f) and (G, g) , we have*

$$d_I(F, G) = d_{FD}(F, G) = d_{FC}(F, G) = d_U(F, G).$$

Previously, only the equality $d_I(F, G) = d_{FD}(F, G)$ was known [3]. We prove Theorem 11 in Section 3, Theorem 12 in Section 4, and Theorem 13 in Section 5.

3 Bi-Lipschitz bounds for Reeb graph distances

This section is devoted to proving Theorem 11. In Section 3.1 we prove $d_I \leq d_{FD} \leq d_{FC} \leq d_U$, in Section 3.2 we prove $d_U \leq 3d_{FD}$, in Section 3.3 we prove $d_U \leq 5d_I$, and in Section 3.4 we prove $d_{FC} \leq 3d_I$. As mentioned before, these inequalities together imply the theorem.

3.1 The inequalities $d_I \leq d_{FD} \leq d_{FC} \leq d_U$

The following lemma and its proof are similar to [4, Proposition 3], but without the assumption that X is compact.

► **Lemma 14.** *Let (F, f) be a Reeb graph induced by a map $\hat{f}: X \rightarrow \mathbb{R}$, let $q: X \rightarrow F$ be the associated quotient map, and suppose $K \subseteq F$ is connected. Then $q^{-1}(K)$ is connected.*

Proof. Suppose the lemma is false. Then $q^{-1}(K) = X_1 \sqcup X_2$, where X_1 and X_2 are nonempty and contained in disjoint open subsets of X . Since F is equipped with the quotient topology of q , $q(X_1)$ and $q(X_2)$ are both open subsets of $q(X_1) \cup q(X_2) = K$. Because K is connected, we must have $q(X_1) \cap q(X_2) \neq \emptyset$.

Let $x \in q(X_1) \cap q(X_2)$, so $V_1 := q^{-1}(x) \cap X_1$ and $V_2 := q^{-1}(x) \cap X_2$ are both nonempty. Since X_1 and X_2 are open and disjoint subsets of $X_1 \cup X_2$, V_1 and V_2 are open and disjoint subsets of $V_1 \cup V_2 = q^{-1}(x)$. But by definition of induced Reeb graph, $q^{-1}(x)$ is connected, so we have a contradiction. \blacktriangleleft

► **Theorem 15.** *Given any two Reeb graphs (F, f) and (G, g) , we have*

$$d_{FC}(F, G) \leq d_U(F, G).$$

Proof. Let X be a topological space with functions $\hat{f}, \hat{g}: X \rightarrow \mathbb{R}$ that induce Reeb graphs (F, f) and (G, g) , respectively, and let $q_F: X \rightarrow F$, $q_G: X \rightarrow G$ denote the corresponding Reeb quotient maps. Suppose $\|\hat{f} - \hat{g}\|_\infty = \delta \geq 0$. For any $\epsilon > 0$, we will construct functions $\phi: F \rightarrow G$ and $\psi: G \rightarrow F$ that form a $(\delta + 2\epsilon)$ -contortion pair; the theorem follows.

Fix $\epsilon > 0$. Pick a discrete subset $S \subseteq F$ containing all the 0-cells of F such that for each 1-cell C and each connected component K of $C \setminus S$, the image $f(K)$ is contained in some interval $[a, b]$ of length $b - a = \epsilon$. Pick a subset $T \subseteq G$ analogously. Define a map $\phi: S \rightarrow G$ by picking an element $\phi(s) \in q_G(q_F^{-1}(s))$ for each $s \in S$.

Let L be the closure of a connected component of $F \setminus S$. Observe that L is contained in a single 1-cell C and is homeomorphic to a closed interval, with endpoints $z, z' \in S$. By our assumptions on S , L is contained in a connected component of $f^{-1}[a, b]$ for some $a < b$ with $b - a = \epsilon$. By Lemma 14, $q_F^{-1}(L)$ is connected, and by continuity of q_G , $J := q_G(q_F^{-1}(L))$ is connected, too. Since J is connected, we can extend ϕ continuously to L by choosing a path from $\phi(z)$ to $\phi(z')$ in J . Moreover, because $\|\hat{f} - \hat{g}\|_\infty = \delta$, we have $J \subseteq g^{-1}[a - \delta, b + \delta] \subseteq g^{-1}(I_{\delta+\epsilon}(f(x)))$. It follows that for every $x \in L$, $\phi(x)$ and $q_G(q_F^{-1}(x))$ are connected in $g^{-1}(I_{\delta+\epsilon}(f(x)))$. We do this for every L as described and get a continuous map $\phi: F \rightarrow G$. Analogously, we get a continuous map $\psi: G \rightarrow F$ such that for any $y \in G$, $\psi(y)$ and $q_F(q_G^{-1}(y))$ are connected in $f^{-1}(I_{\delta+\epsilon}(g(y)))$.

Pick an $x \in L$, where L is as in the previous paragraph, and let $y = \phi(x)$. By construction, $y \in q_G(q_F^{-1}(x'))$ for some $x' \in L$. Thus, $x' \in q_F(q_G^{-1}(y))$, which, as noted, is in the same connected component of $f^{-1}(I_{\delta+\epsilon}(g(y)))$ as $\psi(y)$. But x and x' are connected in $f^{-1}[a, b]$ for some $a < b$ with $b - a = \epsilon$, so it follows that x and $\psi(y)$ are connected in $f^{-1}(I_{\delta+2\epsilon}(g(y)))$. Along with the symmetric statement that follows by a similar argument, this is exactly what is needed for (ϕ, ψ) to be a $(\delta + 2\epsilon)$ -contortion pair. \blacktriangleleft

► **Theorem 16.** *Given any two Reeb graphs (F, f) and (G, g) , we have*

$$d_{FD}(F, G) \leq d_{FC}(F, G).$$

Proof. Suppose $\phi: F \rightarrow G$ and $\psi: G \rightarrow F$ form a δ -contortion pair for some $\delta \geq 0$. Then

$$\|f - g \circ \phi\|_\infty, \|g - f \circ \psi\|_\infty \leq \delta \tag{1}$$

by Remark 9. Let $(x, y), (x', y') \in C(\phi, \psi)$, where $C(\phi, \psi)$ is as in Definition 7. We claim that if x and x' are connected in $f^{-1}[a, b]$ for some $a \leq b$, then y and y' are connected in $g^{-1}[a - \delta, b + \delta]$. Together with the symmetric statement and Equation (1), this is enough to show that (ϕ, ψ) is a δ -distortion pair, from which the lemma follows.

Assume that x and x' lie in the same connected component K of $f^{-1}[a, b]$. We have that $\phi(K)$ is connected, and it follows from Equation (1) that $\phi(K) \subseteq g^{-1}[a - \delta, b + \delta]$. Since $\phi(x), \phi(x') \in \phi(K)$, $\phi(x)$ and $\phi(x')$ are connected in $g^{-1}[a - \delta, b + \delta]$. By definition of $C(\phi, \psi)$, either y is equal to $\phi(x)$, or $y \in \psi^{-1}(x)$. In the latter case, y and $\phi(x)$ are connected in

14:8 Quasi-Universality of Reeb Graph Distances

$g^{-1}(I_\delta(f(x))) \subseteq g^{-1}[a - \delta, b + \delta]$ by definition of δ -contortion. Similarly, y' and $\phi(x')$ are also connected in $g^{-1}[a - \delta, b + \delta]$. Putting everything together, y and y' are connected in $g^{-1}[a - \delta, b + \delta]$, which completes the proof. \blacktriangleleft

► **Theorem 17** ([5, Lemma 8]). *Given any two Reeb graphs (F, f) and (G, g) , we have*

$$d_I(F, G) \leq d_{FD}(F, G).$$

The setting of [5] is slightly different than ours, but the proof of the result carries over.

3.2 Relating universal and functional distortion distance

We denote the connected component of a point p in a space X by $K_p(X)$.

► **Theorem 18.** *Given any two Reeb graphs (F, f) and (G, g) , we have*

$$d_U(F, G) \leq 3d_{FD}(F, G).$$

Proof. Assume that $\phi : F \rightarrow G$ and $\psi : G \rightarrow F$ form a δ -distortion pair. We construct a subspace $Z \subseteq F \times G$ such that the canonical projections $\text{pr}_F : F \times G \rightarrow F$, $\text{pr}_G : F \times G \rightarrow G$ restrict to Reeb quotient maps $q_F : Z \rightarrow F$, $q_G : Z \rightarrow G$ of $f \circ q_F$ and $g \circ q_G$, and $\|f \circ q_F - g \circ q_G\|_\infty \leq 3\delta$, proving that $d_U \leq 3d_{FD}$.

For $x \in F$, let

$$C(x) = K_x(f^{-1}[a, a + 2\delta]),$$

where a is chosen such that $C(x)$ contains $\psi \circ \phi(x)$. By definition of δ -distortion, such an a always exists, though it does not have to be unique. We define $C(y)$ analogously for $y \in G$:

$$C(y) = K_y(g^{-1}[a', a' + 2\delta])$$

for some a' , and $C(y)$ contains $\phi \circ \psi(y)$. Now define

$$Z = \bigcup_{x \in F} C(x) \times \phi(C(x)) \cup \bigcup_{y \in G} \psi(C(y)) \times C(y) \subseteq F \times G$$

and the functions $\hat{f} = f \circ \text{pr}_F$, $\hat{g} = g \circ \text{pr}_G : Z \rightarrow \mathbb{R}$.

To show that $\|\hat{f} - \hat{g}\|_\infty \leq 3\delta$, by symmetry it suffices to show that for every $x \in F$ and every $(z, y) \in C(x) \times \phi(C(x))$ we have $|f(z) - g(y)| \leq 3\delta$. Pick $w \in C(x)$ such that $\phi(w) = y$. We have $|f(z) - f(w)| \leq 2\delta$ by construction of $C(x)$, and $|f(w) - g(y)| \leq \delta$ by definition of δ -distortion. Together, we have $|f(z) - g(y)| \leq 3\delta$ as claimed.

To show that $q_F : Z \rightarrow F$ is surjective, simply observe that for any $x \in F$,

$$(x, \phi(x)) \in C(x) \times \phi(C(x)) \subseteq Z.$$

A similar argument shows that also $q_G : Z \rightarrow G$ is surjective.

It remains to show that the fibers of q_F are connected; by symmetry, the same is then true for q_G as well. The fiber of $z \in F$ is of the form $q_F^{-1}(z) = \{z\} \times G_z \subseteq Z$, where $G_z = q_G(q_F^{-1}(z)) \subseteq G$ is a subspace, homeomorphic to the fiber. Note that G_z has the explicit description

$$G_z = \bigcup_{\substack{x \in F \\ z \in C(x)}} \phi(C(x)) \cup \bigcup_{\substack{y \in G \\ z \in \psi(C(y))}} C(y).$$

Now $\phi(z)$ is contained in any $\phi(C(x))$ with $x \in F$ and $z \in C(x)$, and in any $C(y)$ with $y \in \psi^{-1}(z)$, and each of these subspaces is connected. Thus,

$$G'_z = \bigcup_{\substack{x \in F \\ z \in C(x)}} \phi(C(x)) \cup \bigcup_{\substack{y \in \psi^{-1}(z) \\ z \in \psi(C(y))}} C(y)$$

is connected and contains $\psi^{-1}(z)$ as a subset. Clearly, if $z \in \psi(C(y))$, then $C(y)$ contains an element of $\psi^{-1}(z)$, so $C(y)$ intersects G'_z . As $C(y)$ is connected, it follows that

$$G'_z \cup \bigcup_{\substack{y \in G \\ z \in \psi(C(y))}} C(y) = G_z$$

is connected. ◀

3.3 Relating universal and interleaving distance

► **Lemma 19.** *Let (ϕ, ψ) be a δ -interleaving of (F, f) and (G, g) for some $\delta \geq 0$. If $K \subseteq F$ ($K' \subseteq G$) is connected, then $\Phi(K)$ ($\Psi(K')$) is connected.*

Proof. By continuity of ϕ , $\phi(K) \subseteq \mathcal{U}_\delta G$ is connected. Thus, by Lemma 14,

$$C := q_{\mathcal{U}_\delta G}^{-1}(\phi(K)) \subseteq \mathcal{T}_\delta G$$

is connected. We have that $\Phi(K)$ is exactly the image of C under the projection $\text{pr}_G: \mathcal{T}_\delta G \rightarrow G$. Since this projection is continuous and C is connected, $\Phi(K)$ is connected.

The statement for K' and Ψ follows by symmetry. ◀

► **Theorem 20.** *Given any two Reeb graphs (F, f) and (G, g) , we have*

$$d_U(F, G) \leq 5d_I(F, G).$$

Proof. Let (ϕ, ψ) be a δ -interleaving of (F, f) and (G, g) , so to any $x \in F$, there is associated a subset $\Phi(x) \subseteq G$ that is a connected component of $g^{-1}(I_\delta(f(x)))$. Similarly, for any $y \in G$, $\Psi(y)$ is a connected component of $f^{-1}(I_\delta(g(y)))$. We construct a subspace $Z \subseteq F \times G$ and two functions $\hat{f}, \hat{g}: Z \rightarrow \mathbb{R}$ with $\|\hat{f} - \hat{g}\|_\infty \leq 5\delta$ such that the canonical projections $\text{pr}_F: F \times G \rightarrow F$, $\text{pr}_G: F \times G \rightarrow G$ restrict to Reeb quotient maps $q_F: Z \rightarrow F$ of \hat{f} and $q_G: Z \rightarrow G$ of \hat{g} , proving that $d_U \leq 5d_I$.

For $x \in F$ and $y \in G$, let

$$C(x) = K_x(f^{-1}(I_{2\delta}(f(x)))), \quad C(y) = K_y(g^{-1}(I_{2\delta}(g(y))))$$

and let

$$Z = \bigcup_{x \in F} C(x) \times \Phi(C(x)) \cup \bigcup_{y \in G} \Psi(C(y)) \times C(y) \subseteq F \times G.$$

To show that $\|\hat{f} - \hat{g}\|_\infty \leq 5\delta$, by symmetry it suffices to show that for every $x \in F$ and every $(z, y) \in C(x) \times \Phi(C(x))$ we have $|f(z) - g(y)| \leq 5\delta$. Pick $w \in C(x)$ such that $y \in \Phi(w)$. We have $|f(z) - f(w)| \leq 4\delta$ by construction of $C(x)$, and $|f(w) - g(y)| \leq \delta$ by definition of δ -interleaving. Together, we have $|f(z) - g(y)| \leq 5\delta$ as claimed.

To show that $q_F: Z \rightarrow F$ is surjective, simply observe that for any $x \in F$ and $y \in \Phi(x)$,

$$(x, y) \in C(x) \times \Phi(C(x)) \subseteq Z.$$

A similar argument shows that also $q_G: Z \rightarrow G$ is surjective.

14:10 Quasi-Universality of Reeb Graph Distances

It remains to show that the fibers of q_F are connected; by symmetry, the same is then true for q_G as well. The fiber of $z \in F$ is of the form $q_F^{-1}(z) = \{z\} \times G_z \subseteq Z$, where $G_z = q_G(q_F^{-1}(z)) \subseteq G$ is a subspace, homeomorphic to the fiber. Note that G_z has the explicit description

$$G_z = \bigcup_{\substack{x \in F \\ z \in C(x)}} \Phi(C(x)) \cup \bigcup_{\substack{y \in G \\ z \in \Psi(y)}} C(y).$$

Clearly, $\Phi(z)$ is contained in any $\Phi(C(x))$ with $x \in F$ and $z \in C(x)$. In addition, $\Phi(z) \subseteq C(y)$ for any $y \in G$ such that $z \in \Psi(y)$, as $\Phi(\Psi(y)) \subseteq C(y)$ by definition of interleaving. By Lemma 19, $\Phi(C(x))$ is connected. Thus,

$$G'_z = \bigcup_{\substack{x \in F \\ z \in C(x)}} \Phi(C(x)) \cup \bigcup_{\substack{y \in G \\ z \in \Psi(y)}} C(y) \subseteq G_z$$

is connected, since it is a union of connected sets that all contain $\Psi(z)$. To complete the proof that G_z is connected, it suffices to show that $C(y)$ intersects G'_z for all $y \in G$ such that $z \in \Psi(C(y))$. To see this, observe that there is a $w \in C(y)$ such that $z \in \Psi(w)$, and $w \in C(w) \subseteq G'_z$. ◀

3.4 Relating functional contortion and interleaving distance

► **Theorem 21.** *Given any two Reeb graphs (F, f) and (G, g) , we have*

$$d_{FC}(F, G) \leq 3d_I(F, G).$$

Proof. Let (ϕ, ψ) be a δ -interleaving of (F, f) and (G, g) for some $\delta \geq 0$. We will show that for an arbitrary $\epsilon > 0$, there are $\mu: F \rightarrow G$ and $\nu: G \rightarrow F$ with functional contortion $3\delta + 3\epsilon$.

Pick a discrete subset $S \subset F$ containing all the 0-cells of F such that for each 1-cell C and connected component I of $C \setminus S$, the interval $f(I)$ has length less than ϵ . Pick a discrete subset $T \subset G$ with the same properties.

For each $x \in S$, pick an arbitrary $y \in \Phi(x)$ and define $\mu(x) = y$. Similarly, for each $y \in T$, let $\nu(y) \in \Psi(y)$. Let I be a connected component of $F \setminus S$. Observe that I is contained in a single 1-cell C , and that its closure \bar{I} contains two points $z, z' \in S$. By our assumptions on I , \bar{I} is contained in a connected component of $f^{-1}[a, b]$ for some $a < b$ with $b - a = \epsilon$. By Lemma 19, it follows that $\Phi(\bar{I})$ is contained in a connected component K of $g^{-1}[a - \delta, b + \delta]$. We can therefore use a path from $\mu(z)$ to $\mu(z')$ in K to extend μ continuously to \bar{I} ; see Figure 3. This implies that for all $x \in \bar{I}$, $\mu(x)$ and $\Phi(x)$ are both contained in K and thus also in the same component of $g^{-1}(I_{\delta+\epsilon}(f(x)))$, as

$$g^{-1}[a - \delta, b + \delta] \subseteq g^{-1}(I_{\delta+\epsilon}(f(x))).$$

We do the same for all connected components of $F \setminus S$ and define ν similarly on $G \setminus T$.

Let $\delta' = \delta + \epsilon$. We now prove that (μ, ν) is a $3\delta'$ -contortion pair. By symmetry, it is enough to show that for any $x \in F$, x and $\nu(\mu(x))$ are connected in $f^{-1}(I_{3\delta'}(g(\mu(x))))$. The δ -interleaving (Φ, Ψ) induces a δ' -interleaving (Φ', Ψ') canonically: For $x \in F$, $\Phi'(x)$ is the connected component of $g^{-1}(I_{\delta'}(f(x)))$ containing $\Phi(x)$ as a subset, and $\Psi'(y)$ is defined similarly for $y \in G$. We observed that $\mu(x)$ and $\Phi(x)$ are connected in $g^{-1}(I_{\delta'}(f(x)))$, so

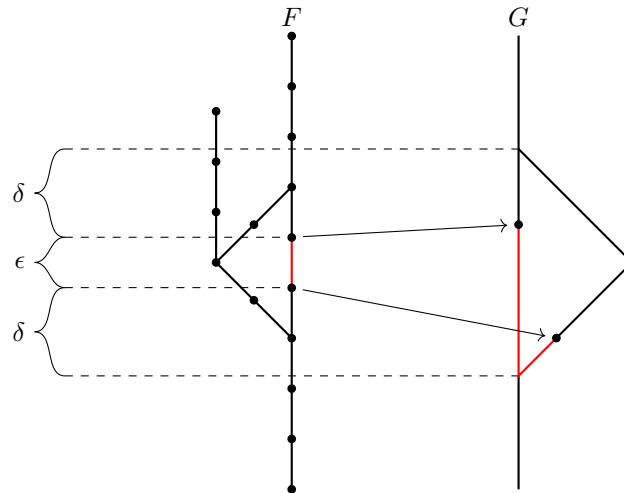


Figure 3 Construction of a functional contortion pair for Theorem 21. The points of S are shown as black dots. The arrows and the red segments in F and G show μ applied to two points in S and how we can extend μ to the segment between the points.

$\mu(x) \in \Phi'(x)$. Similarly, $\nu(\mu(x)) \in \Psi'(\mu(x))$. Putting the two together, we get $\nu(\mu(x)) \in \Psi'(\Phi'(x))$. By definition of interleaving, we have $\Psi'(\Phi'(x)) \subseteq K_x(f^{-1}(I_{2\delta'}(f(x))))$. Since $|f(x) - g(\mu(x))| \leq \delta'$, we have

$$f^{-1}(I_{2\delta'}(f(x))) \subseteq f^{-1}(I_{3\delta'}(g(\mu(x))))$$

so

$$\nu(\mu(x)) \in \Psi'(\Phi'(x)) \subseteq K_x(f^{-1}(I_{3\delta'}(g(\mu(x))))),$$

which is what we wanted to prove. ◀

4 Contour trees

Proof of Theorem 12. We know $d_{FC}(F, G) \leq d_U(F, G)$ by Theorem 15, so it remains to prove $d_{FC}(F, G) \geq d_U(F, G)$.

Assume that there is a δ -contortion (ϕ, ψ) between F and G . We construct a subspace $Z \subseteq F \times G$ and two functions $\hat{f}, \hat{g}: Z \rightarrow \mathbb{R}$ with $\|\hat{f} - \hat{g}\|_\infty \leq \delta$ such that the canonical projections $\text{pr}_F: F \times G \rightarrow F$, $\text{pr}_G: F \times G \rightarrow G$ restrict to Reeb quotient maps $q_F: Z \rightarrow F$ of \hat{f} and $q_G: Z \rightarrow G$ of \hat{g} , proving that $d_{FC}(F, G) \geq d_U(F, G)$.

Let $x \neq x' \in F$, and let $\rho: [0, 1] \rightarrow F$ be an injective path from x to x' . Since F is a contour tree and therefore a contractible 1-dimensional CW complex, the image of this path is uniquely determined, so we can define $B(x, x') = \text{im}(\rho)$. Observe that $z \in B(x, x') \setminus \{x, x'\}$ if and only if x and x' are in different connected components of $F \setminus \{z\}$. $B(y, y')$ for $y, y' \in G$ is defined similarly.

Let $Z \subseteq F \times G$ be given by

$$Z = \left[\bigcup_{x \in F, y \in \psi^{-1}(x)} \{x\} \times B(\phi(x), y) \right] \cup \left[\bigcup_{y \in G, x \in \phi^{-1}(y)} B(\psi(y), x) \times \{y\} \right].$$

14:12 Quasi-Universality of Reeb Graph Distances

To show that $\|\hat{f} - \hat{g}\|_\infty \leq \delta$, by symmetry it suffices to show that for every $x \in F$, $y \in \psi^{-1}(x)$ and $y' \in B(\phi(x), y)$, $|f(x) - g(y')| \leq \delta$. But by definition of δ -contortion, $\phi(x)$ and y are connected in $g^{-1}[f(x) - \delta, f(x) + \delta]$, so $B(\phi(x), y) \subseteq g^{-1}[f(x) - \delta, f(x) + \delta]$, and the statement follows.

For any $x \in F$, we have

$$(x, \phi(x)) \in B(\psi \circ \phi(x), x) \times \{\phi(x)\} \subseteq Z,$$

and it follows immediately that $q_F: Z \rightarrow F$ is surjective, and similarly for $q_G: Z \rightarrow G$.

It remains to show that the fibers of q_F and q_G are connected. By symmetry, we only need to prove this for q_F . The fiber of $x \in F$ is of the form $q_F^{-1}(x) = \{x\} \times G_x \subseteq Z$, where $G_x = q_G(q_F^{-1}(x)) \subseteq G$ is a subspace, homeomorphic to the fiber. To follow the arguments that follow, we suggest keeping an eye on Figure 4. Let

$$U_x = \{\phi(x)\} \cup \bigcup_{y \in \psi^{-1}(x)} B(\phi(x), y) \subseteq G.$$

(Including $\{\phi(x)\}$ only makes a difference if $\phi^{-1}(x)$ is empty.) Since U_x is a union of connected sets intersecting in the point $\phi(x)$, U_x is connected. Moreover, $U_x \subseteq G_x$ by construction, as we have already observed that $(x, \phi(x)) \in q_F^{-1}(x)$.

Let $D_x = \{\phi(x)\} \cup \psi^{-1}(x)$. Note that $D_x \subseteq U_x$. Let $y \in G_x \setminus U_x$. Then there is an $x' \in \phi^{-1}(y)$ such that $x \in B(\psi(y), x')$. Equivalently, x' and $\psi(y)$ are in different connected components F_1 and F_2 , respectively, of $F \setminus \{x\}$. (Since $y \notin D_x$, neither x' nor $\psi(y)$ is equal to x .) Since D_x is closed, so is $\phi^{-1}(D_x)$. This means that we can pick an $x'' \in \phi^{-1}(D_x)$ such that

$$B := B(x', x'') \setminus \{x''\}$$

does not intersect $\phi^{-1}(D_x)$. It follows that $B \subseteq F_1$, since $x \notin B$. It also follows that $\psi \circ \phi(B) \subseteq F_2$, since $x \notin \psi \circ \phi(B)$ and $\psi(y) \in \psi \circ \phi(B)$. Thus, for all $z \in B$, we have $x \in B(\psi \circ \phi(z), z)$; i.e.,

$$(x, \phi(z)) \in B(\psi \circ \phi(z), z) \times \{\phi(z)\} \subseteq Z,$$

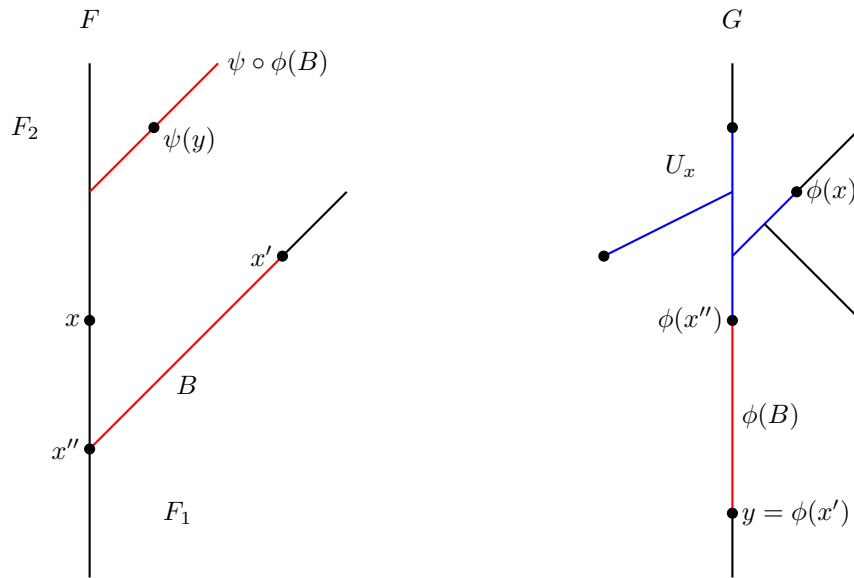
so $\phi(B) \subseteq G_x$. This means that there is a path in G_x from $y = \phi(x')$ to $\phi(x'') \in U_x$. Since y was an arbitrary point in G_x and U_x is connected, it follows that G_x is connected. \blacktriangleleft

5 Merge trees

In this section, we focus on merge trees, which are a special case of contour trees that also arise from the connected components of the sublevel set filtration of a function.

The merge trees obtained this way carry a function that is unbounded above, and they are characterized by the property that the canonical map from the merge tree to the Reeb graph of its epigraph is an isomorphism [10]. Our definition is more general and also admits bounded functions, and in Section 5.1 we develop an analogous characterization for these general merge trees via the property that said canonical map is an embedding. Relating this property to our definition is not straightforward, and we defer the proofs to [1, Appendix C].

Our goal in Section 5.2 is to prove that the interleaving distance for merge trees is universal. By Theorem 12, it suffices to construct a δ -contortion pair from a δ -interleaving of merge trees. Summarizing the idea for the simpler special case of a merge tree G unbounded above, the key insight behind the proof is that the δ -smoothing of G is isomorphic to an upward δ -shift of G . Composing the interleaving morphisms with the isomorphisms obtained this way yields the desired δ -contortion pair in the unbounded case.



■ **Figure 4** Illustration of constructions used to prove connectedness of fibers of q_F .

5.1 The epigraph and merge trees

We formally characterize merge trees using a construction based on *epigraphs*, as previously suggested by Morozov et al. [10].

► **Definition 22** (Epigraph). *Let $f: X \rightarrow \mathbb{R}$ be a continuous function. We define the epigraph of f as the space $\mathcal{E}X := X \times [0, \infty)$ equipped with the function $\mathcal{E}f: \mathcal{E}X \rightarrow \mathbb{R}, (p, t) \mapsto f(p) + t$.*

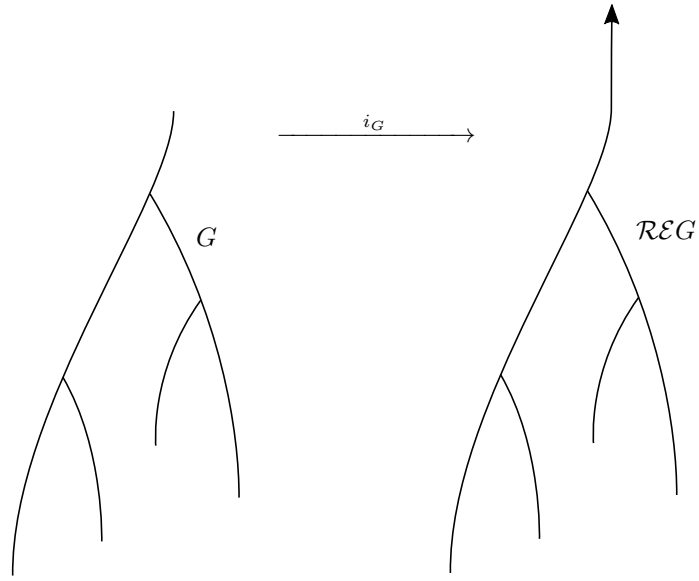
While this is not the usual definition of the epigraph $\{(p, y) \in X \times \mathbb{R} \mid f(p) \leq y\}$, we note that the map $\mathcal{E}f: \mathcal{E}X \rightarrow \mathbb{R}$ and the projection of the ordinary epigraph to the second component are isomorphic as \mathbb{R} -spaces. Our definition has the benefit that we have the strict equality $\delta + \mathcal{E}f = \mathcal{E}(\delta + f): \mathcal{E}X \rightarrow \mathbb{R}$ for any $\delta \in \mathbb{R}$.

Now suppose that (G, g) is a Reeb graph, and let $m := \sup_{p \in G} g(p) \in (-\infty, \infty]$. We now define the map $\tilde{\rho}_G: \mathcal{E}G \rightarrow \mathcal{E}G, (p, t) \mapsto (p, \min\{t, m - g(p)\})$, which makes the diagram

$$\begin{array}{ccc}
 (\mathcal{E}g)^{-1}(-\infty, m] & \xlongequal{\quad} & (\mathcal{E}g)^{-1}(-\infty, m] \\
 \downarrow & & \downarrow \\
 \mathcal{E}G & \xrightarrow{\tilde{\rho}_G} & \mathcal{E}G \\
 \mathcal{E}g \downarrow & & \downarrow \mathcal{E}g \\
 \mathbb{R} & \xrightarrow{\min\{-, m\}} & \mathbb{R}
 \end{array}$$

commute. We state the following immediate consequence of this definition.

► **Lemma 23.** *For each $t \in \mathbb{R}$ the map $\tilde{\rho}_G: \mathcal{E}G \rightarrow \mathcal{E}G$ restricts to a homeomorphism between the fibers $(\mathcal{E}g)^{-1}(t)$ and $(\mathcal{E}g)^{-1}(\min\{t, m\})$.*



■ **Figure 5** The embedding $i_G: G \rightarrow \mathcal{R}\mathcal{E}G$ of a merge tree G into its unbounded variant $\mathcal{R}\mathcal{E}G$.

By the universal property of the quotient topology, there is a unique continuous map $\mathcal{R}\tilde{\rho}_G: \mathcal{R}\mathcal{E}G \rightarrow \mathcal{R}\mathcal{E}G$ making the following diagram commute:

$$\begin{array}{ccc}
 \mathcal{E}G & \xrightarrow{\tilde{\rho}_G} & \mathcal{E}G \\
 \downarrow q_{\mathcal{R}\mathcal{E}G} & & \downarrow q_{\mathcal{R}\mathcal{E}G} \\
 \mathcal{R}\mathcal{E}G & \xrightarrow{\mathcal{R}\tilde{\rho}_G} & \mathcal{R}\mathcal{E}G \\
 \downarrow \mathcal{R}\mathcal{E}g & & \downarrow \mathcal{R}\mathcal{E}g \\
 \mathbb{R} & \xrightarrow{\min\{-,m\}} & \mathbb{R}
 \end{array} \tag{2}$$

► **Corollary 24.** For each $t \in \mathbb{R}$ the map $\mathcal{R}\tilde{\rho}_G: \mathcal{R}\mathcal{E}G \rightarrow \mathcal{R}\mathcal{E}G$ restricts to a bijection between the fibers $(\mathcal{R}\mathcal{E}g)^{-1}(t)$ and $(\mathcal{R}\mathcal{E}g)^{-1}(\min\{t, m\})$.

Let $\kappa_X: X \rightarrow X \times [0, \infty) = \mathcal{E}X$, $p \mapsto (p, 0)$ denote the natural embedding of X into the epigraph of f . We state several properties (see [1, Appendix C] for the proofs) of the map

$$i_G: G \xrightarrow{\kappa_G} \mathcal{E}G \xrightarrow{q_{\mathcal{R}\mathcal{E}G}} \mathcal{R}\mathcal{E}G.$$

► **Lemma 25.** The images of the maps $i_G: G \rightarrow \mathcal{R}\mathcal{E}G$ and $\mathcal{R}\tilde{\rho}_G: \mathcal{R}\mathcal{E}G \rightarrow \mathcal{R}\mathcal{E}G$ are identical.

► **Lemma 26.** A Reeb graph (G, g) is a merge tree iff the map $i_G: G \rightarrow \mathcal{R}\mathcal{E}G$ is an embedding.

Now suppose that (G, g) is a merge tree. The composite map

$$i_G: G \xrightarrow{\kappa_G} \mathcal{E}G \xrightarrow{q_{\mathcal{R}\mathcal{E}G}} \mathcal{R}\mathcal{E}G$$

is non-surjective iff $g: G \rightarrow \mathbb{R}$ is bounded above. We define $\rho_G: \mathcal{R}\mathcal{E}G \rightarrow G$ to be the unique continuous map – which exists by Lemmas 25 and 26 – making the diagram

$$\begin{array}{ccc}
 G & \xleftarrow{\rho_G} & \mathcal{R}\mathcal{E}G \\
 \kappa_G \downarrow & & \downarrow \mathcal{R}\tilde{\rho}_G \\
 \mathcal{E}G & \xrightarrow{q_{\mathcal{R}\mathcal{E}G}} & \mathcal{R}\mathcal{E}G
 \end{array} \tag{3}$$

commute. As an immediate corollary of Corollary 24, we obtain the following observation.

► **Corollary 27.** *For each $t \in \mathbb{R}$, the map $\rho_G: \mathcal{R}\mathcal{E}G \rightarrow G$ restricts to a bijection between the fibers $(\mathcal{R}\mathcal{E}g)^{-1}(t)$ and $g^{-1}(\min\{t, m\})$.*

5.2 Interleavings, contortions, and merge trees

Let $f: X \rightarrow \mathbb{R}$ be an arbitrary continuous function and let $\delta \geq 0$. We define the map

$$\kappa_X^\delta: \mathcal{T}_\delta X \rightarrow \mathcal{E}X, (p, t) \mapsto (p, t + \delta),$$

which makes the diagram

$$\begin{array}{ccc}
 \mathcal{T}_\delta X & \xrightarrow{\kappa_X^\delta} & \mathcal{E}X \\
 \mathcal{T}_\delta f \downarrow & & \downarrow \mathcal{E}f \\
 \mathbb{R} & \xrightarrow{(-)+\delta} & \mathbb{R}
 \end{array}$$

commute. Now let (G, g) be a merge tree. By the universal property of the quotient topology, there is a unique continuous map $\mathcal{R}\kappa_G^\delta: \mathcal{U}_\delta G \rightarrow \mathcal{R}\mathcal{E}G$ making the diagram

$$\begin{array}{ccccc}
 \mathcal{T}_\delta G & \xrightarrow{\kappa_G^\delta} & \mathcal{E}G & & \\
 \downarrow \mathcal{T}_\delta g & \searrow q_{\mathcal{U}_\delta G} & \downarrow \mathcal{E}g & \searrow q_{\mathcal{R}\mathcal{E}G} & \\
 & \mathcal{U}_\delta G & \xrightarrow{\mathcal{R}\kappa_G^\delta} & \mathcal{R}\mathcal{E}G & \\
 & \swarrow \mathcal{U}_\delta g & \downarrow & \swarrow \mathcal{R}\mathcal{E}g & \\
 \mathbb{R} & \xrightarrow{(-)+\delta} & \mathbb{R} & &
 \end{array} \tag{4}$$

commute.

► **Lemma 28.** *The map $\mathcal{R}\kappa_G^\delta: \mathcal{U}_\delta G \rightarrow \mathcal{R}\mathcal{E}G$ is injective.*

As in the previous subsection, let $m := \sup_{p \in G} g(p) \in (-\infty, \infty]$, let $p \in G$, and let $\delta' = \delta$ or $\delta' \in [-\delta, \delta]$ if $g(p) = m$.

► **Lemma 29.** *The composite map*

$$G \times [-\delta, \delta] = \mathcal{T}_\delta G \xrightarrow{q_{\mathcal{U}_\delta G}} \mathcal{U}_\delta G \xrightarrow{\mathcal{R}\kappa_G^\delta} \mathcal{R}\mathcal{E}G \xrightarrow{\rho_G} G$$

maps $(p, -\delta')$ to p .

14:16 Quasi-Universality of Reeb Graph Distances

Proof of Theorem 13. Suppose (F, f) and (G, g) are merge trees and that

$$\phi: F \rightarrow \mathcal{U}_\delta G \quad \text{and} \quad \psi: G \rightarrow \mathcal{U}_\delta F$$

form a δ -interleaving (of Reeb graphs). We show that the composite maps

$$\tilde{\phi}: F \xrightarrow{\phi} \mathcal{U}_\delta G \xrightarrow{\mathcal{R}\kappa_G^\delta} \mathcal{R}\mathcal{E}G \xrightarrow{\rho_G} G, \quad \tilde{\psi}: G \xrightarrow{\psi} \mathcal{U}_\delta F \xrightarrow{\mathcal{R}\kappa_F^\delta} \mathcal{R}\mathcal{E}F \xrightarrow{\rho_F} F$$

form a δ -contortion pair. Together with Theorem 12, this proves the claim.

Let $x \in F$ and let $y \in \tilde{\psi}^{-1}(x)$. We have to show that y and $\tilde{\phi}(x)$ are connected in $g^{-1}(I_\delta(f(x)))$. By the symmetry of Definition 8, this is also sufficient. By the commutativity of the lower parallelogram in (4) the value of $\mathcal{R}\kappa_F^\delta(\psi(y))$ under $\mathcal{R}\mathcal{E}f$ is

$$(\mathcal{U}_\delta f)(\psi(y)) + \delta = g(y) + \delta.$$

In conjunction with the commutativity of (3) and the lower parallelogram in (2) we obtain

$$f(x) = (f \circ \tilde{\psi})(y) = \min\{g(y) + \delta, m'\},$$

where $m' := \sup_{p \in F} f(p)$, and hence

$$f(x) - g(y) = \min\{g(y) + \delta, m'\} - g(y) = \min\{\delta, m' - g(y)\}.$$

Moreover, $g(y) = (\mathcal{U}_\delta f)(\psi(y)) \leq m' + \delta$, so in conjunction with Lemma 29 we obtain that

$$G \times [-\delta, \delta] = \mathcal{T}_\delta G \xrightarrow{q_{\mathcal{U}_\delta G}} \mathcal{U}_\delta G \xrightarrow{\mathcal{R}\kappa_G^\delta} \mathcal{R}\mathcal{E}G \xrightarrow{\rho_G} G$$

maps $(x, g(y) - f(x))$ to x . Thus, the composite map

$$\mathcal{U}_\delta G \xrightarrow{\mathcal{R}\kappa_G^\delta} \mathcal{R}\mathcal{E}G \xrightarrow{\rho_G} G$$

maps both $q_{\mathcal{U}_\delta F}(x, g(y) - f(x))$ and $\psi(y)$ to x . By Lemma 28 and Corollary 27 this implies

$$q_{\mathcal{U}_\delta F}(x, g(y) - f(x)) = \psi(y).$$

Completely analogously we obtain that $q_{\mathcal{U}_\delta G}(\tilde{\phi}(x), f(x) - (g \circ \tilde{\phi})(x)) = \phi(x)$. Thus, y and $\tilde{\phi}(x)$ are connected in $g^{-1}(I_{2\delta}(g(y)))$ by Definition 6. It remains to show that y and $\tilde{\phi}(x)$ are connected in $g^{-1}(I_\delta(f(x)))$. To this end, let $t := \min\{g(y) + 2\delta, m\}$, where $m := \sup_{p \in G} g(p)$.

▷ **Claim 30.** We have $(g \circ \tilde{\phi})(x) = t$.

Proof. We first consider the case $(f \circ \tilde{\psi})(y) = f(x) < m'$. In this case, we have $f(x) = (f \circ \tilde{\psi})(y) = g(y) + \delta$ and thus $(g \circ \tilde{\phi})(x) = t$. Now suppose $f(x) = m'$. Since $\tilde{\phi}(x) \in g^{-1}(I_{2\delta}(g(y)))$, we must have $(g \circ \tilde{\phi})(x) \leq t$. Now suppose $(g \circ \tilde{\phi})(x) < t \leq m$. Then $(g \circ \tilde{\phi})(x) = f(x) + \delta = m' + \delta$. In particular, we have $m' + \delta < t \leq m$. Now let $s \in (m' + \delta, m)$. Then we have $(\mathcal{U}_\delta f)^{-1}(s) = \emptyset$ while $g^{-1}(s) \neq \emptyset$, a contradiction to the existence of the map $\psi|_{g^{-1}(s)}: g^{-1}(s) \rightarrow (\mathcal{U}_\delta f)^{-1}(s)$. ◁

We obtain the connectivity of y and $\tilde{\phi}(x)$ in $g^{-1}(I_\delta(f(x)))$ from their connectivity in $g^{-1}(I_{2\delta}(g(y)))$ by define a retraction $\sigma: g^{-1}(I_{2\delta}(g(y))) \rightarrow g^{-1}(t)$ as a composition of maps

$$\begin{array}{c}
 g^{-1}(I_{2\delta}(g(y))) \\
 \downarrow \\
 g^{-1}(-\infty, g(y) + 2\delta] \\
 \downarrow \kappa_G \\
 (\mathcal{E}g)^{-1}(-\infty, g(y) + 2\delta] \\
 \downarrow \tilde{\sigma} \\
 (\mathcal{E}g)^{-1}[t, g(y) + 2\delta] \\
 \downarrow q_{\mathcal{R}\mathcal{E}G} \\
 (\mathcal{R}\mathcal{E}g)^{-1}[t, g(y) + 2\delta] \\
 \downarrow \rho_G \\
 g^{-1}(t),
 \end{array}$$

where

$$\tilde{\sigma}: (\mathcal{E}g)^{-1}(-\infty, g(y) + 2\delta] \rightarrow (\mathcal{E}g)^{-1}[t, g(y) + 2\delta], (p, s) \mapsto (p, \max\{s, t - g(p)\}).$$

By the definition of $\tilde{\sigma}$ the map $\sigma: g^{-1}(I_{2\delta}(g(y))) \rightarrow g^{-1}(t)$ is indeed a retraction. As $\tilde{\phi}(x) \in g^{-1}(t)$ by Claim 30 the points $\sigma(y)$ and $\tilde{\phi}(x)$ are connected in the fiber $g^{-1}(t)$. Since the fibers of g are discrete, this implies that $\sigma(y) = \tilde{\phi}(x)$. Defining the path

$$\gamma: [0, 1] \rightarrow \mathcal{E}G, s \mapsto (y, s(t - g(y)))$$

the composition

$$[0, 1] \xrightarrow{\gamma} \mathcal{E}G \xrightarrow{q_{\mathcal{R}\mathcal{E}G}} \mathcal{R}\mathcal{E}G \xrightarrow{\rho_G} G$$

yields a path from y to $\sigma(y) = \tilde{\phi}(x)$ in $g^{-1}(I_\delta(f(x)))$. ◀

References

- 1 Ulrich Bauer, Håvard Bakke Bjerkevik, and Benedikt Fluhr. Quasi-universality of Reeb graph distances. Preprint, 2021. [arXiv:2112.00720](#).
- 2 Ulrich Bauer, Xiaoyin Ge, and Yusu Wang. Measuring distance between Reeb graphs. In *Computational geometry (SoCG'14)*, pages 464–473. ACM, New York, 2014.
- 3 Ulrich Bauer, Xiaoyin Ge, and Yusu Wang. Measuring distance between Reeb graphs. Extended version of conference paper, 2016. [arXiv:1307.2839v2](#).
- 4 Ulrich Bauer, Claudia Landi, and Facundo Mémoli. The Reeb graph edit distance is universal. *Found. Comput. Math.*, 21(5):1441–1464, 2021. [doi:10.1007/s10208-020-09488-3](#).
- 5 Ulrich Bauer, Elizabeth Munch, and Yusu Wang. Strong equivalence of the interleaving and functional distortion metrics for Reeb graphs. In *31st International Symposium on Computational Geometry*, volume 34 of *LIPICs. Leibniz Int. Proc. Inform.*, pages 461–475. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2015.
- 6 Robert Cardona, Justin Curry, Tung Lam, and Michael Lesnick. The universal ℓ^p -metric on merge trees. Preprint, 2021. [arXiv:2112.12165](#).

- 7 Michele d'Amico, Patrizio Frosini, and Claudia Landi. Natural pseudo-distance and optimal matching between reduced size functions. *Acta Appl. Math.*, 109(2):527–554, 2010. doi:10.1007/s10440-008-9332-1.
- 8 Vin de Silva, Elizabeth Munch, and Amit Patel. Categorified Reeb graphs. *Discrete Comput. Geom.*, 55(4):854–906, 2016. doi:10.1007/s00454-016-9763-9.
- 9 Masaki Hilaga, Yoshihisa Shinagawa, Taku Komura, and Toshiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In Lynn Pocock, editor, *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001, Los Angeles, California, USA, August 12-17, 2001*, pages 203–212. ACM, 2001. doi:10.1145/383259.383282.
- 10 Dmitriy Morozov, Kenes Beketayev, and Gunther Weber. Interleaving distance between merge trees. Presented at TopoInVis'13. Manuscript, 2013. URL: <https://www.mrzv.org/publications/interleaving-distance-merge-trees/>.
- 11 Georges Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *C. R. Acad. Sci. Paris*, 222:847–849, 1946.
- 12 Luis N. Scoccola. *Locally persistent categories and metric properties of interleaving distances*. PhD thesis, The University of Western Ontario, 2020. URL: <https://ir.lib.uwo.ca/etd/7119>.
- 13 Yoshihisa Shinagawa and Toshiyasu L. Kunii. Constructing a Reeb graph automatically from cross sections. *IEEE Computer Graphics and Applications*, 11(6):44–51, 1991. doi:10.1109/38.103393.
- 14 Gurjeet Singh, Facundo Mémoli, and Gunnar E. Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In Mario Botsch, Renato Pajarola, Baoquan Chen, and Matthias Zwicker, editors, *4th Symposium on Point Based Graphics, PBG@Eurographics 2007, Prague, Czech Republic, September 2-3, 2007*, pages 91–100. Eurographics Association, 2007. doi:10.2312/SPBG/SPBG07/091-100.

Gromov Hyperbolicity, Geodesic Defect, and Apparent Pairs in Vietoris–Rips Filtrations

Ulrich Bauer   

Department of Mathematics and Munich Data Science Institute,
Technische Universität München, Germany

Fabian Roll   

Department of Mathematics, Technische Universität München, Germany

Abstract

Motivated by computational aspects of persistent homology for Vietoris–Rips filtrations, we generalize a result of Eliyahu Rips on the contractibility of Vietoris–Rips complexes of geodesic spaces for a suitable parameter depending on the hyperbolicity of the space. We consider the notion of geodesic defect to extend this result to general metric spaces in a way that is also compatible with the filtration. We further show that for finite tree metrics the Vietoris–Rips complexes collapse to their corresponding subforests. We relate our result to modern computational methods by showing that these collapses are induced by the apparent pairs gradient, which is used as an algorithmic optimization in Ripser, explaining its particularly strong performance on tree-like metric data.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Mathematics of computing → Trees; Theory of computation → Computational geometry

Keywords and phrases Vietoris–Rips complexes, persistent homology, discrete Morse theory, apparent pairs, hyperbolicity, geodesic defect, Ripser

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.15

Related Version *Extended Version*: <https://arxiv.org/abs/2112.06781>

Funding This research has been supported by the DFG Collaborative Research Center SFB/TRR 109 *Discretization in Geometry and Dynamics*.

Acknowledgements We thank Michael Bleher, Lukas Hahn, and Andreas Ott for stimulating discussions about applications of Vietoris–Rips complexes to the topological study of coronavirus evolution motivating our interest in the persistence of tree metrics, the organizers and participants of the AATRN Vietoris–Rips online seminar for sparking our interest in the Contractibility Lemma, and the anonymous reviewers for valuable feedback.

1 Introduction

The *Vietoris–Rips* complex is a fundamental construction in algebraic, geometric, and applied topology. For a metric space X and a threshold $t > 0$, it is defined as the simplicial complex consisting of nonempty and finite subsets of X with diameter at most t :

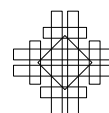
$$\text{Rips}_t(X) = \{\emptyset \neq S \subseteq X \mid S \text{ finite, } \text{diam } S \leq t\}.$$

First introduced by Vietoris [27] in order to make homology applicable to general compact metric spaces, it has also found important applications in geometric group theory [16] and topological data analysis [26]. The role of the threshold in these three application areas is notably different. The homology theory defined by Vietoris arises in the limit $t \rightarrow 0$. In contrast, the key applications in geometric group theory rely on the fact that the Vietoris–Rips complex of a hyperbolic geodesic space is contractible for a sufficiently large threshold.



© Ulrich Bauer and Fabian Roll;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 15; pp. 15:1–15:15
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



15:2 Gromov Hyperbolicity, Geodesic Defect, and Apparent Pairs in Rips Filtrations

This observation, originally due to Rips and first published in Gromov’s seminal paper on hyperbolic groups [16], is a fundamental result about the topology of Vietoris–Rips complexes and plays a central role in the theory of hyperbolic groups.

► **Lemma 1** (Contractibility Lemma; Rips, Gromov [16]). *Let X be a δ -hyperbolic geodesic metric space. Then the complex $\text{Rips}_t(X)$ is contractible for every $t > 0$ with $t \geq 4\delta$.*

Here, a metric space (X, d) is called *geodesic* if for any two points $x, y \in X$ there exists an isometric map $[0, d(x, y)] \rightarrow X$ such that $0 \mapsto x$ and $d(x, y) \mapsto y$, and it is called *δ -hyperbolic* (in the sense of Gromov [16]) for $\delta \geq 0$ if for any four points $w, x, y, z \in X$ we have

$$d(w, x) + d(y, z) \leq \max\{d(w, y) + d(x, z), d(w, z) + d(x, y)\} + 2\delta. \quad (1)$$

Finally, in applications of Vietoris–Rips complexes to topological data analysis, one is typically interested in the *persistent homology* of the entire filtration of complexes for all possible thresholds. A notable difference to the classical applications is that the metric spaces under consideration are typically finite, and in particular not geodesic. This motivates the interest in a meaningful generalization of the Contractibility Lemma to finite metric spaces. Based on the notion of a *discretely geodesic space* defined by Lang [22], which is the natural setting for hyperbolic groups, and motivated by techniques used in that paper, we consider the following quantitative geometric property (called *ν -almost geodesic* in [10, p. 271]).

► **Definition 2.** *A metric space X is ν -geodesic if for all $x, y \in X$ and $r, s \geq 0$ with $r + s = d(x, y)$ there exists a point $z \in X$ with $d(x, z) \leq r + \nu$ and $d(y, z) \leq s + \nu$. The geodesic defect of X , denoted by $\nu(X)$, is the infimum over all ν such that X is ν -geodesic.*

Our first main result is a generalization of the Contractibility Lemma that also applies to non-geodesic spaces using our notion of geodesic defect, and further produces collapses that are compatible with the Vietoris–Rips filtration above the collapsibility threshold.

► **Theorem 3.** *Let X be a finite δ -hyperbolic ν -geodesic metric space. Then there exists a discrete gradient that induces, for every $u > t \geq 4\delta + 2\nu$, a sequence of collapses*

$$\text{Rips}_u(X) \searrow \text{Rips}_t(X) \searrow \{*\}.$$

► **Example 4.** An important special case is given by a finite *tree metric space* (V, d) , where V is the vertex set of a positively weighted tree $T = (V, E)$, and where the edge weights are taken as lengths and d is the associated path length metric, i.e., for two points $x, y \in V$ their distance is the infimum total weight of any path starting in x and ending in y . The geodesic defect is $\nu(V) = \frac{1}{2} \max_{e \in E} l(e)$, where $l(e)$ is the length of the edge e . Moreover, (V, d) is 0-hyperbolic (see [13, Theorems 3.38 and 3.40] for a characterization of 0-hyperbolic spaces).

This example is of particular relevance in the context of evolutionary biology, where persistent Vietoris–Rips homology has been successfully applied to identify recombinations and recurrent mutations [11, 24, 8]. The metrics arising as genetic distances of aligned RNA or DNA sequences are typically very similar to trees, capturing the phylogeny of the evolution. This motivates our interest in the particular case of tree metrics. These metric spaces are known to have acyclic Vietoris–Rips homology in degree > 0 , and so any homology is an indication of some evolutionarily relevant phenomenon.

Our second main result is a strengthened version of Theorem 3 for the special case of tree metric spaces that connects the collapses of the Vietoris–Rips filtration to the construction of *apparent pairs*, which play an important role as a computational shortcut in the software

Ripser [6]. This result depends on a particular ordering of the vertices: we say that a total order of V is *compatible* with the tree T if it extends the unique tree partial order resulting from choosing some arbitrary root vertex as the minimal element.

► **Theorem 5.** *Let V be a finite tree metric space for a weighted tree $T = (V, E)$, whose vertices are totally ordered in a compatible way. Then the apparent pairs gradient for the lexicographically refined Vietoris–Rips filtration induces a sequence of collapses*

$$\text{Rips}_u(V) \searrow \text{Rips}_t(V) \searrow T_t$$

for every $u > t > 0$ such that no edge $e \in E$ has length $l(e) \in (t, u]$, where T_t is the subforest with vertices V and all edges of E with length at most t . In particular, the persistent homology of the Vietoris–Rips filtration is trivial in degree > 0 .

In the special case of trees with unit edge length, the proofs in [1, Proposition 2.2] and [2, Proposition 3] are similar in spirit to our proof of Theorem 25, which is based on discrete Morse theory. Related results about implications of the geometry of a metric space on the homotopy types of the associated Vietoris–Rips complexes can be found in [3, 4, 23].

► **Remark 6.** Given a vertex order \leq , the lexicographic order on simplices for the reverse vertex order \geq coincides with the reverse colexicographic order for the original order \leq , which is used for computations in Ripser. As a consequence, when the input is a tree metric with the points ordered in reverse order of the distances to some arbitrarily chosen root, then Ripser will identify all non-tree simplices in apparent pairs, requiring not a single column operation to compute its trivial persistent homology. In practice, we observe that on data that is almost tree-like, such as genetic evolution distances, Ripser exhibits exceptionally good computational performance. The results of this paper provide a partial geometric explanation for this behavior and yield a heuristic for preprocessing tree-like data by sorting the points to speed up the computation in such cases. In the application to the study of SARS-CoV-2 described in [8], ordering the genome sequences in reverse chronological order, as an approximation of the reverse tree order for the phylogenetic tree, lead to a huge performance improvement, bringing down the computation time for the persistence barcode from a full day to about 2 minutes.

2 Preliminaries

2.1 Discrete Morse theory and the apparent pairs gradient

A *simplicial complex* K on a vertex set $\text{Vert } K$ is a collection of nonempty finite subsets of $\text{Vert } K$ such that for any set $\sigma \in K$ and any nonempty subset $\rho \subseteq \sigma$ one has $\rho \in K$. A set $\sigma \in K$ is called a *simplex*, and $\dim \sigma = \text{card } \sigma - 1$ is its *dimension*. Moreover, ρ is said to be a *face of* σ and σ a *coface of* ρ . If $\dim \rho = \dim \sigma - 1$, then we call ρ a *facet of* σ and σ a *cofacet of* ρ . The *star of* σ , $\text{St } \sigma$, is the set of cofaces of σ in K , and the *closure of* σ , $\text{Cl } \sigma$, is the set of its faces. For a subset $E \subseteq K$, we write $\text{St } E = \bigcup_{\sigma \in E} \text{St } \sigma$.

Generalizing the ideas of Forman [14], a function $f: K \rightarrow \mathbb{R}$ is a *discrete Morse function* [7, 15] if f is monotonic, i.e., for any $\sigma, \tau \in K$ with $\sigma \subseteq \tau$ we have $f(\sigma) \leq f(\tau)$, and there exists a partition of K into intervals $[\rho, \phi] = \{\psi \in K \mid \rho \subseteq \psi \subseteq \phi\}$ in the face poset such that $f(\sigma) = f(\tau)$ for any $\sigma \subseteq \tau$ if and only if σ and τ belong to a common interval in the partition. The collection of *regular intervals*, $[\rho, \phi]$ with $\rho \neq \phi$, is called the *discrete gradient of* f , and any singleton interval $[\sigma, \sigma]$, as well as the corresponding simplex σ , is called *critical*.

15:4 Gromov Hyperbolicity, Geodesic Defect, and Apparent Pairs in Rips Filtrations

► **Proposition 7** (Hersh [18, Lemma 4.1]; Jonsson [20, Lemma 4.2]). *Let K be a finite simplicial complex, and $\{K_\alpha\}_{\alpha \in A}$ a set of subcomplexes covering K , each equipped with a discrete gradient V_α , such that for any simplex of K*

- *there is a unique minimal subcomplex K_α containing that simplex, and*
- *the simplex is critical for the discrete gradients of all other such subcomplexes.*

Then the regular intervals in the V_α are disjoint, and their union is a discrete gradient on K .

An *elementary collapse* $K \searrow K \setminus \{\sigma, \tau\}$ is the removal of a pair of simplices, where σ is a facet of τ , with τ the unique proper coface of σ . A *collapse* $K \searrow L$ onto a subcomplex L is a sequence of elementary collapses starting in K and ending in L . An elementary collapse can be realized continuously by a strong deformation retraction and therefore collapses preserve the homotopy type. A discrete gradient can encode a collapse.

► **Proposition 8** (Forman [14]; see also [21, Theorem 10.9]). *Let K be a finite simplicial complex and let $L \subseteq K$ be a subcomplex. Assume that V is a discrete gradient on K such that the complement $K \setminus L$ is the union of intervals in V . Then there exists a collapse $K \searrow L$.*

Let $f: K \rightarrow \mathbb{R}$ be a monotonic function. Assume that the vertices of K are totally ordered. The *f -lexicographic order* is the total order \leq_f on K given by ordering the simplices

- by their value under f ,
- then by dimension,
- then by the lexicographic order induced by the total vertex order.

We call a pair (σ, τ) of simplices in K a *zero persistence pair* if $f(\sigma) = f(\tau)$. An *apparent pair* (σ, τ) with respect to the f -lexicographic order is a pair of simplices in K such that σ is the maximal facet of τ , and τ is the minimal cofacet of σ . The collection of apparent pairs forms a discrete gradient [6, Lemma 3.5], called the *apparent pairs gradient*.

Assume that K is finite and $f: K \rightarrow \mathbb{R}$ a discrete Morse function with discrete gradient V . Refine V to another discrete gradient

$$\tilde{V} = \{(\psi \setminus \{v\}, \psi \cup \{v\}) \mid \psi \in [\rho, \phi] \in V, v = \min(\phi \setminus \rho)\}$$

by doing a minimal vertex refinement on each interval.

► **Lemma 9.** *The zero persistence apparent pairs with respect to the f -lexicographic order are precisely the gradient pairs of \tilde{V} .*

Proof. Let (σ, τ) be a zero persistence apparent pair. Then $f(\sigma) = f(\tau)$, and σ and τ are contained in the same regular interval $I = [\rho, \phi]$ of V . Let v be the minimal vertex in $\phi \setminus \rho$. By assumption, σ is the maximal facet of τ in I , and τ is the minimal cofacet of σ . Hence, σ is lexicographically maximal among all facets of τ in I , and τ is lexicographically minimal under all cofacets of σ in I . By the assumption that (σ, τ) forms an apparent pair, we cannot have $v \in \sigma$, as otherwise $\tau \setminus \{v\}$ would be a larger facet of τ than σ . Similarly, we cannot have $v \notin \tau$, as otherwise $\sigma \cup \{v\}$ would be a smaller cofacet of σ than τ . This means that $\tau = \sigma \cup \{v\}$ and therefore $\{\sigma, \tau\} \in \tilde{V}$.

Conversely, assume that $\{\sigma, \tau\} \in \tilde{V}$ holds. Consider the interval $I = [\rho, \phi]$ of V with $\{\sigma, \tau\} \subseteq I$ and let v be the minimal vertex in $\phi \setminus \rho$. By construction of \tilde{V} , $\sigma = \tau \setminus \{v\}$ is the lexicographically maximal facet of τ in I and $\tau = \sigma \cup \{v\}$ is the lexicographically minimal cofacet of σ in I . Therefore, (σ, τ) is a zero persistence apparent pair. ◀

2.2 Rips' Contractibility Lemma via the injective hull

In this section, we recall some known facts about embeddings of metric spaces into their injective hull. We adapt these results using our notion of geodesic defect to prove a version of the Contractibility Lemma for finite δ -hyperbolic ν -geodesic metric spaces, following [25].

Let Y be a metric space. The *Čech complex* of a subspace $X \subseteq Y$ for radius $r > 0$ is the nerve of the collection of closed balls in Y with radius r centered at points in X :

$$\check{C}ech_r(X, Y) = \{ \emptyset \neq S \subseteq X \mid S \text{ finite, } \bigcap_{x \in S} D_r(x) \neq \emptyset \},$$

where $D_r(x) = \{y \in Y \mid d(x, y) \leq r\}$ denotes the closed ball in Y of radius r centered at x .

A metric space is *hyperconvex* [12] if it is geodesic and if any collection of closed balls has the Helly property, i.e., if any two of these balls have a nonempty intersection, then all balls have a nonempty intersection. The following lemma is a direct consequence of this definition.

► **Lemma 10.** *If Y is hyperconvex and $X \subseteq Y$ is a subspace, then $\check{C}ech_r(X, Y) = \text{Rips}_{2r}(X)$.*

Let X be a metric space. We describe its *injective hull* $E(X)$, following Lang [22]. A function $f: X \rightarrow \mathbb{R}$ with $f(x) + f(y) \geq d(x, y)$ for all $x, y \in X$ is *extremal* if $f(x) = \sup_{y \in X} (d(x, y) - f(y))$ for every $x \in X$. The difference between any two extremal functions turns out to be bounded, and so we can equip the set $E(X)$ of extremal functions with the metric induced by the supremum norm, i.e., $d(f, g) = \sup_{x \in X} |f(x) - g(x)|$. We define an isometric embedding $e: X \rightarrow E(X)$ by $y \mapsto d_y$, where $d_y(x) = d(y, x)$.

► **Remark 11.** $E(X)$ is a hyperconvex space. In particular, $E(X)$ is contractible, and nonempty intersections of closed metric balls are contractible [22, 19]. Moreover, nonempty intersections of open metric balls are also contractible [25, Proposition 2.8 and Lemma 2.15].

The following theorem is essentially due to Lang [22]. Originally, it has been stated for a special case, but the proof applies verbatim to the below statement involving our notion of the geodesic defect, which indeed provided the motivation for our definition. Note that the definition of δ -hyperbolic used in [22] differs from the one used here by a factor of 2.

► **Proposition 12** (Lang [22, Proposition 1.3]). *Let X be a δ -hyperbolic ν -geodesic metric space. Then the injective hull $E(X)$ is δ -hyperbolic, and every point in $E(X)$ has distance at most $2\delta + \nu$ to $e(X)$.*

Now we prove a generalization of the Contractibility Lemma using the injective hull analogously to the proof for geodesic spaces in [25, Corollary 8.4].

► **Theorem 13.** *Let X be a finite δ -hyperbolic ν -geodesic metric space. Then the complex $\text{Rips}_t(X)$ is contractible for every $t \geq 4\delta + 2\nu$.*

Proof. By Proposition 12, we know that for $r > \frac{t}{2} \geq 2\delta + \nu$ the collection of open balls with radius r centered at the points in $e(X)$ covers $E(X)$. By finiteness of X , there exists an $r > \frac{t}{2}$ such that the nerve of this cover is isomorphic to $\check{C}ech_{\frac{t}{2}}(e(X), E(X))$. As e is an isometric embedding, Lemma 10, Remark 11, and the Nerve Theorem [17, Section 4.G] imply

$$\text{Rips}_t(X) = \text{Rips}_t(e(X)) = \check{C}ech_{\frac{t}{2}}(e(X), E(X)) \simeq E(X) \simeq *. \quad \blacktriangleleft$$

3 Filtered collapsibility of Vietoris–Rips complexes

In this section, we revisit the original proof of the Contractibility Lemma in [16], adapted to the language of discrete Morse theory [14]. Focusing on the finite case, which also constitutes the key part of the original proof, we extend the statement beyond geodesic spaces using our notion of geodesic defect, strengthen the assertion of contractibility to collapsibility, and further extend the result to become compatible with the Vietoris–Rips filtration.

► **Theorem 14.** *Let X be a finite δ -hyperbolic ν -geodesic metric space. Then for every $t \geq 4\delta + 2\nu$ there exists a discrete gradient that induces a collapse $\text{Rips}_t(X) \searrow \{*\}$.*

Proof. Without loss of generality, assume that $\delta > 0$; if X is 0-hyperbolic, then it is also ϵ -hyperbolic for any $\epsilon > 0$, and for sufficiently small $\epsilon > 0$ we have $\text{Rips}_{4\epsilon+2\nu}(X) = \text{Rips}_{2\nu}(X)$.

Choose a reference point $p \in X$ and order the points according to their distance to p , choosing a total order $p = x_1 < \dots < x_n$ on X such that $x_i < x_j$ implies $d(x_i, p) \leq d(x_j, p)$. Let $t \geq 4\delta + 2\nu$ and consider the filtration

$$\{p\} = K_1 \subseteq \dots \subseteq K_n = \text{Rips}_t(X),$$

where $K_i = \text{Rips}_t(X_i)$ for $X_i := \{x_1, \dots, x_i\}$. We prove that for $i \in \{2, \dots, n\}$ there exists a discrete gradient V_i on K_i inducing a collapse $K_i \searrow K_{i-1}$.

First assume $d(x_i, p) < t$. Then for any vertex x_k of K_i we have $k \leq i$ and $d(x_k, p) \leq d(x_i, p) < t$, so K_i is a simplicial cone with apex p . Pairing the simplices containing p with those not containing p , we obtain a discrete gradient inducing a collapse $K_i \searrow K_{i-1}$:

$$V_i = \{(\sigma \setminus \{p\}, \sigma \cup \{p\}) \mid \sigma \in K_i \setminus K_{i-1}\}.$$

Now assume $d(x_i, p) \geq t$. We show that there exists a point $z \in X_{i-1}$ such that for every simplex $\sigma \in K_i \setminus K_{i-1}$, the union $\sigma \cup \{z\}$ is also a simplex in $K_i \setminus K_{i-1}$. To this end, we show that any vertex y of σ has distance $d(y, z) \leq t$ to z . For $r = d(x_i, p) - 2\delta - \nu$ and $s = 2\delta + \nu$ we have $r + s = d(x_i, p)$, and therefore, by the assumption that X is a ν -geodesic space, there exists a point $z \in X$ with $d(z, p) \leq r + \nu = d(x_i, p) - 2\delta$, implying $z < x_i$, and $d(z, x_i) \leq s + \nu = 2\delta + 2\nu$. By assumption $t \geq 4\delta + 2\nu$, and thus we get $d(z, x_i) \leq t - 2\delta$. Note that $y \in X_i$ implies $d(y, p) \leq d(x_i, p)$, and $y, x_i \in \sigma$ implies $d(y, x_i) \leq \text{diam } \sigma \leq t$. The four-point condition (1) now yields

$$\begin{aligned} d(y, z) &\leq \max\{d(y, x_i) + d(z, p), d(y, p) + d(z, x_i)\} + 2\delta - d(x_i, p) \\ &= \max\{\underbrace{d(y, x_i)}_{\leq t} + \underbrace{d(z, p) - d(x_i, p)}_{\leq -2\delta}, \underbrace{d(y, p) - d(x_i, p)}_{\leq 0} + \underbrace{d(z, x_i)}_{\leq t-2\delta}\} + 2\delta \leq t. \end{aligned} \quad (2)$$

Similarly to the above, pairing the simplices containing z with those not containing z yields a discrete gradient inducing a collapse $K_i \searrow K_{i-1}$:

$$V_i = \{(\sigma \setminus \{z\}, \sigma \cup \{z\}) \mid \sigma \in K_i \setminus K_{i-1}\}.$$

Finally, by Proposition 7, the union $V = \bigcup_i V_i$ is a discrete gradient on $\text{Rips}_t(X)$ and by Proposition 8 it induces a collapse $\text{Rips}_t(X) \searrow \{p\}$. ◀

► **Remark 15.** For a simplicial complex K , a particular type of simplicial collapse called an *elementary strong collapse* from K to $K \setminus \text{St } v$ is defined in [5] for the case where the link of the vertex v is a simplicial cone. The proof of Theorem 14 actually shows that for $t \geq 4\delta + 2\nu$ there exists a sequence of elementary strong collapses from $\text{Rips}_t(X)$ to $\{*\}$.

We can now extend the proof strategy of Theorem 14 to obtain a filtration-compatible strengthening of the Contractibility Lemma.

Proof of Theorem 3. As in the proof of Theorem 14, we can assume that $\delta > 0$, and order the points in X according to their distance to a chosen reference point $p = x_1 < \dots < x_n$.

As X is finite, we can enumerate the values of pairwise distances by $0 = r_0 < \dots < r_l$. For every $r_m > 4\delta + 2\nu$ we construct a discrete gradient W_m inducing a collapse $\text{Rips}_{r_m}(X) \searrow \text{Rips}_{r_{m-1}}(X)$. This will prove the theorem, because it follows from Theorem 14 that there exists a discrete gradient V that induces a collapse $\text{Rips}_{4\delta+2\nu}(X) \searrow \{*\}$, and an application of Proposition 7 assembles these gradients into a single gradient $W = V \cup \bigcup_m W_m$ on $\text{Cl}(X)$ inducing collapses $\text{Rips}_u(X) \searrow \text{Rips}_t(X) \searrow \{*\}$ for every $u > t \geq 4\delta + 2\nu$.

Let m be arbitrary such that $r_m > 4\delta + 2\nu$. Consider the filtration

$$\text{Rips}_{r_{m-1}}(X) = K_1 \subseteq \dots \subseteq K_n = \text{Rips}_{r_m}(X),$$

where $K_i = \text{Rips}_{r_{m-1}}(X) \cup \text{Rips}_{r_m}(X_i)$ for $X_i := \{x_1, \dots, x_i\}$. We prove that for $i \in \{2, \dots, n\}$ there exists a discrete gradient V_i on K_i inducing a collapse $K_i \searrow K_{i-1}$. Note that $K_i \setminus K_{i-1}$ consists of all simplices of diameter r_m that contain x_i as the maximal vertex.

First assume $d(x_i, p) < r_m$. Let $\sigma \in K_i \setminus K_{i-1}$. As x_i is the maximal vertex of σ , we have $d(v, p) \leq d(x_i, p) < r_m$ for all $v \in \sigma$. Since σ has diameter r_m , this implies that $\sigma \cup \{p\}$ also has diameter r_m . Moreover, this implies that there exists an edge $e \subseteq \sigma \setminus \{p\} \subseteq \sigma$ not containing p with $\text{diam } e = r_m$. Therefore, $\sigma \setminus \{p\}$ also has diameter r_m . As $p < x_i$, both simplices $\sigma \setminus \{p\}$ and $\sigma \cup \{p\}$ contain x_i as the maximal vertex and are thus contained in $K_i \setminus K_{i-1}$. Pairing the simplices containing p with those not containing p , we obtain a discrete gradient inducing a collapse $K_i \searrow K_{i-1}$:

$$V_i = \{(\sigma \setminus \{p\}, \sigma \cup \{p\}) \mid \sigma \in K_i \setminus K_{i-1}\}.$$

Now assume $d(x_i, p) \geq r_m$. We show that there exists a point $z \in X_{i-1}$ such that for every simplex $\sigma \in K_i \setminus K_{i-1}$, the simplices $\sigma \setminus \{z\}$ and $\sigma \cup \{z\}$ are also contained in $K_i \setminus K_{i-1}$. To this end, we show first that any vertex y of σ has distance $d(y, z) \leq r_m$ to z . As in the proof of Theorem 14, there exists a point $z \in X$ with $d(z, p) \leq d(x_i, p) - 2\delta$, implying $z < x_i$, and $d(z, x_i) \leq 2\delta + 2\nu$. By assumption $r_m > 4\delta + 2\nu$, and thus we get $d(z, x_i) < r_m - 2\delta$. Similar to Equation (2), we have the following estimate

$$d(y, z) \leq \max\left\{\underbrace{d(y, x_i)}_{\leq r_m} + \underbrace{d(z, p) - d(x_i, p)}_{\leq -2\delta}, \underbrace{d(y, p) - d(x_i, p)}_{\leq 0} + \underbrace{d(z, x_i)}_{< r_m - 2\delta}\right\} + 2\delta \leq r_m,$$

and if $d(y, x_i) < r_m$, then $d(y, z) < r_m$. Hence, $\text{diam}(\sigma \cup \{z\}) = r_m$, and $\text{diam } \sigma = r_m$ implies $\text{diam } \sigma \setminus \{z\} = r_m$, by an argument similar to the above. As $z < x_i$, both simplices $\sigma \setminus \{z\}$ and $\sigma \cup \{z\}$ contain x_i as the maximal vertex and are thus contained in $K_i \setminus K_{i-1}$. Pairing the simplices containing z with those not containing z , we obtain a discrete gradient inducing a collapse $K_i \searrow K_{i-1}$:

$$V_i = \{(\sigma \setminus \{z\}, \sigma \cup \{z\}) \mid \sigma \in K_i \setminus K_{i-1}\}.$$

By Proposition 7 the union $W_m = \bigcup V_i$ is a discrete gradient on $\text{Rips}_{r_m}(X)$, and by Proposition 8 it induces a collapse $\text{Rips}_{r_m}(X) \searrow \text{Rips}_{r_{m-1}}(X)$. ◀

4 Collapsing Vietoris–Rips complexes of trees by apparent pairs

In this section, we analyze the Vietoris–Rips filtration of a tree metric space (V, d) for a positively weighted finite tree $T = (V, E)$, with the goal of proving the collapses in Theorem 5 using the apparent pairs gradient. To this end, we introduce two other discrete gradients: the *canonical gradient*, which is independent of any choices, and the *perturbed gradient*, which coarsens the canonical gradient and can be interpreted as a gradient that arises through a symbolic perturbation of the edge lengths. We then show that the intervals in the perturbed gradient are refined by apparent pairs of the lexicographically refined Vietoris–Rips filtration, with respect to a particular total order on the vertices.

We write $D_r(x) = \{y \in V \mid d(x, y) \leq r\}$ and $S_r(x) = \{y \in V \mid d(x, y) = r\}$.

► **Lemma 16.** *Let $x, y \in V$ be two distinct points at distance $d(x, y) = r$. Then we have $\text{diam } D_r(x) \cap D_r(y) = r$. Furthermore, if $a, b \in D_r(x) \cap D_r(y)$ are points with $d(a, b) = r$, then these points are contained in the union $S_r(x) \cup S_r(y)$.*

Proof. We start by showing the first claim. Let $a, b \in D_r(x) \cap D_r(y)$ be any two points. We show that $d(a, b) \leq r$ holds, implying $\text{diam } D_r(x) \cap D_r(y) \leq r$. Because $x, y \in D_r(x) \cap D_r(y)$ we also have $\text{diam } D_r(x) \cap D_r(y) \geq r$, proving equality.

Write $[n] = \{1, \dots, n\}$ and let $\gamma: ([n], \{\{i, i+1\} \mid i \in [n-1]\}) \rightarrow T$ be the unique shortest path $x \rightsquigarrow y$. Moreover, let Ψ_a and Ψ_b be the unique shortest paths $x \rightsquigarrow a$ and $x \rightsquigarrow b$, respectively. Consider the largest numbers $t_a, t_b \in [n]$ with $\gamma(t_a) = \Psi_a(t_a)$ and $\gamma(t_b) = \Psi_b(t_b)$ and assume without loss of generality $t_a \leq t_b$. Note that the unique shortest path $a \rightsquigarrow b$ is then given by the concatenation $a \rightsquigarrow \gamma(t_a) \rightsquigarrow \gamma(t_b) \rightsquigarrow b$, where $\gamma(t_a) \rightsquigarrow \gamma(t_b)$ is the restricted path $\gamma|_{[t_a, t_b]}$. By assumption, we have $d(a, y) \leq r$ and this implies the inequality

$$d(a, \gamma(t_a)) + d(\gamma(t_a), y) = d(a, y) \leq r = d(x, y) = d(x, \gamma(t_a)) + d(\gamma(t_a), y),$$

which is equivalent to $d(a, \gamma(t_a)) \leq d(x, \gamma(t_a))$. Similarly, the assumption $d(x, b) \leq r$ implies $d(\gamma(t_b), b) \leq d(\gamma(t_b), y)$. Thus, the distance $d(a, b)$ satisfies

$$\begin{aligned} d(a, b) &= d(a, \gamma(t_a)) + d(\gamma(t_a), \gamma(t_b)) + d(\gamma(t_b), b) \\ &\leq d(x, \gamma(t_a)) + d(\gamma(t_a), \gamma(t_b)) + d(\gamma(t_b), y) = d(x, y) = r, \end{aligned} \quad (3)$$

which finishes the proof of the first claim.

We now show the second claim; assume $d(a, b) = r$. From the inequalities (3) and $d(a, \gamma(t_a)) \leq d(x, \gamma(t_a))$, $d(\gamma(t_b), b) \leq d(\gamma(t_b), y)$ together with the assumption $d(a, b) = r$, we deduce the equalities $d(a, \gamma(t_a)) = d(x, \gamma(t_a))$ and $d(\gamma(t_b), b) = d(\gamma(t_b), y)$. Hence,

$$d(a, y) = d(a, \gamma(t_a)) + d(\gamma(t_a), y) = d(x, \gamma(t_a)) + d(\gamma(t_a), y) = d(x, y) = r$$

and similarly $d(x, b) = r$, proving the second claim. ◀

Enumerate the values of pairwise distances by $0 = r_0 < \dots < r_l = \text{diam } V$. Let $K_m := \text{Rips}_{r_{m-1}}(V) \cup T_{r_m}$. We show that the complement $C_m := \text{Rips}_{r_m}(V) \setminus K_m$ is the set of all cofaces of non-tree edges of length r_m . We further show that it is partitioned into regular intervals in the face poset, and that this constitutes a discrete gradient.

► **Lemma 17.** *Every edge $e \in \text{Rips}_{r_m}(V) \setminus \text{Rips}_{r_{m-1}}(V)$ is contained in a unique maximal simplex $\Delta_e \in \text{Rips}_{r_m}(V) \setminus \text{Rips}_{r_{m-1}}(V)$. Moreover, if e is a tree edge of length r_m , then $\Delta_e = e$, and if $e \in C_m$, then $\Delta_e \in C_m$ and $e \subsetneq \Delta_e$.*

Proof. By definition, e corresponds to two points $x, y \in V$ at distance $d(x, y) = r_m$. If e is contained in the simplex $\Delta \in \text{Rips}_{r_m}(V)$, then the points in Δ lie in the intersection $D_{r_m}(x) \cap D_{r_m}(y)$, which has diameter r_m by Lemma 16. Hence, the maximal simplex Δ_e is spanned by all the points in $D_{r_m}(x) \cap D_{r_m}(y)$.

If e is a tree edge of length r_m , then this intersection only contains x and y , and hence $\Delta_e = e$. If $e \in C_m$, then this intersection contains at least one vertex different from x and y that lies on the unique shortest path $x \rightsquigarrow y$. This implies $e \subsetneq \Delta_e$. ◀

For every maximal simplex $\Delta \in C_m \subseteq \text{Rips}_{r_m}(V)$, we write E_Δ for the set of edges $e \in C_m$ with $\Delta_e = \Delta$. Note that E_Δ is the set of non-tree edges of length r_m contained in Δ .

4.1 Generic tree metrics

Before dealing with the general case, let us focus on the special case where the metric space (V, d) is *generic*, meaning that the pairwise distances are distinct. In this case, Lemma 17 implies that the diameter function $\text{diam}: \text{Cl}(V) \rightarrow \mathbb{R}$ is a discrete Morse function, defined on the full simplicial complex on V , with discrete gradient

$$\{[e, \Delta_e] \mid \text{non-tree edge } e \subseteq \text{Cl}(V)\},$$

which we call the *generic gradient*, and only the vertices V and the tree edges E are critical. Together with Proposition 8, this yields the following theorem.

► **Theorem 18.** *If the tree metric space (V, d) is generic, then the generic gradient induces, for every $m \in \{1, \dots, l\}$, a sequence of collapses*

$$\text{Rips}_{r_m}(V) \searrow (\text{Rips}_{r_{m-1}}(V) \cup T_{r_m}) \searrow T_{r_m}.$$

Moreover, it follows from Lemma 9 that for the Vietoris–Rips filtration, refined lexicographically with respect to an arbitrary total order on the vertices, the zero persistence apparent pairs refine the generic gradient, and therefore also induce the above collapses.

► **Theorem 19.** *If the tree metric space (V, d) is generic, then the apparent pairs gradient induces, for every $m \in \{1, \dots, l\}$, a sequence of collapses*

$$\text{Rips}_{r_m}(V) \searrow (\text{Rips}_{r_{m-1}}(V) \cup T_{r_m}) \searrow T_{r_m}.$$

4.2 Arbitrary tree metrics

We now turn to the general case, where Lemma 9 is not directly applicable anymore, as the diameter function is not necessarily a discrete Morse function. Nevertheless, we show that Theorem 19 is still true without the genericity assumption, if the vertices V are ordered in a compatible way. Let Δ be a maximal simplex $\Delta \in C_m \subseteq \text{Rips}_{r_m}(V)$.

► **Lemma 20.** *We have $\text{St } E_\Delta = C_m \cap \text{Cl } \Delta$.*

Proof. The inclusion $\text{St } E_\Delta \subseteq C_m \cap \text{Cl } \Delta$ holds by definition of E_Δ . To show the inclusion $\text{St } E_\Delta \supseteq C_m \cap \text{Cl } \Delta$, let $\sigma \in C_m \cap \text{Cl } \Delta$ be any simplex. As the Vietoris–Rips complex is a clique complex, there exists an edge $e \subseteq \sigma \subseteq \Delta$ with $\text{diam } e = r_m$. By Lemma 17, this edge can not be a tree edge end hence $e \in C_m$. Therefore, $e \in E_\Delta$ and $\sigma \in \text{St } e \subseteq \text{St } E_\Delta$. ◀

► **Lemma 21.** *If two distinct maximal simplices $\Delta, \Delta' \in C_m = \text{Rips}_{r_m}(V) \setminus K_m$ intersect in a common face $\Delta \cap \Delta'$, then this face is contained in K_m .*

15:10 Gromov Hyperbolicity, Geodesic Defect, and Apparent Pairs in Rips Filtrations

Proof. Assume for a contradiction that $\emptyset \neq \Delta \cap \Delta' \notin K_m$, implying $\Delta \cap \Delta' \in C_m$. By Lemma 20, there exists an edge $e \in E_\Delta \subseteq C_m$ with $e \subseteq \Delta \cap \Delta'$, and therefore $\Delta = \Delta'$ by uniqueness of the maximal simplex containing e (Lemma 17), a contradiction. \blacktriangleleft

We denote by L_Δ the set of all vertices of Δ that are not contained in any edge in E_Δ .

► **Lemma 22.** *Let $e = \{u, w\} \in E_\Delta$ be an edge. Then any point $x \in V \setminus \{u, w\}$ on the unique shortest path $u \rightsquigarrow w$ of length r_m in T is contained in L_Δ . In particular, L_Δ is nonempty.*

Proof. By assumption, we have $d(u, x) < r_m$, $d(w, x) < r_m$ and $d(u, w) = r_m$. Therefore, $\text{diam}\{u, w, x\} = r_m$ and $x \in \{u, w, x\} \subseteq \Delta_e = \Delta$. Assume for a contradiction that x is contained in an edge in E_Δ . Then it follows from Lemma 16 that we have $d(u, x) = r_m$ or $d(w, x) = r_m$, contradicting the above. We conclude that $x \in L_\Delta$. \blacktriangleleft

4.2.1 The canonical gradient

We now describe a discrete gradient that is compatible with the diameter function and induces the same collapses as in Theorem 18 even if the tree metric is not generic. This construction is *canonical* in the sense that it does not depend on the choice of an order on the vertices, in contrast to the subsequent constructions.

► **Lemma 23.** *For any two edges $f, e \in E_\Delta$ and any vertex $v \in f$ there exists a vertex $z \in e$ such that $\{v, z\} \in E_\Delta$ is an edge in E_Δ .*

Proof. Let $f = \{v, w\}$, $e = \{x, y\}$; note that $d(v, w) = d(x, y) = r_m$. Since f and e are both contained in the maximal simplex Δ , we have $v, w \in D_{r_m}(x) \cap D_{r_m}(y)$. Both $\{v, x\}$ and $\{v, y\}$ are contained in $\{v, x, y\} \subseteq \Delta$ and Lemma 16 implies that at least one of these two edges is contained in $\text{Rips}_{r_m}(V) \setminus \text{Rips}_{r_{m-1}}(V)$; call this edge e_v . It follows from Lemma 17 that e_v is not a tree edge, and therefore $e_v \in E_\Delta$. \blacktriangleleft

► **Lemma 24.** *The set $\text{St } E_\Delta = C_m \cap \text{Cl } \Delta$ is partitioned by the intervals*

$$W_\Delta = \{[\cup S, (\cup S) \cup L_\Delta] \mid \emptyset \neq S \subseteq E_\Delta\}, \quad (4)$$

and these form a discrete gradient on $\text{Cl } \Delta$ inducing a collapse $\text{Cl } \Delta \searrow (K_m \cap \text{Cl } \Delta)$.

Proof. The intervals in W_Δ are disjoint and contained in $\text{St } E_\Delta$ by construction. They are regular, because L_Δ is nonempty (by Lemma 22). By Proposition 8, it remains to show that the intervals in W_Δ partition $\text{St } E_\Delta = \text{Cl } \Delta \setminus (K_m \cap \text{Cl } \Delta)$ and that W_Δ is a discrete gradient.

To show the first claim, it suffices to prove that any simplex $\sigma \in \text{St } E_\Delta$ is contained in a regular interval of W_Δ . Consider the simplex $\tau = \sigma \setminus L_\Delta \subseteq \sigma$. As $\sigma \in \text{St } E_\Delta$, there exists an edge $e \in E_\Delta$ with $e \subseteq \sigma$. By the definition of L_Δ , we have $e \subseteq \sigma \setminus L_\Delta = \tau$. Any other vertex $v \in \tau \setminus e$ is also contained in one of the edges E_Δ . By Lemma 23, there exists an edge $e_v = \{v, w\} \in E_\Delta$, where $w \in e$. Then $\tau = e \cup \bigcup_{v \in \tau \setminus e} e_v$ and $\sigma \in [\tau, \tau \cup L_\Delta] \in W_\Delta$.

The second claim now follows from the observation that the function

$$\sigma \mapsto \begin{cases} \dim(\sigma \cup L_\Delta) & \sigma \in \text{St } E_\Delta \\ \dim \sigma & \sigma \notin \text{St } E_\Delta \end{cases}$$

is a discrete Morse function with discrete gradient W_Δ . \blacktriangleleft

Consider the union $W_m = \bigcup_\Delta W_\Delta$, where Δ runs over all maximal simplices in C_m and W_Δ is as in (4). We call $W = \bigcup_m W_m$ the *canonical gradient*.

► **Theorem 25.** *The canonical gradient is a discrete gradient on $\text{Cl}(V)$. For every $m \in \{1, \dots, l\}$, it induces a sequence of collapses*

$$\text{Rips}_{r_m}(V) \searrow \text{Rips}_{r_{m-1}}(V) \cup T_{r_m} \searrow T_{r_m}.$$

Proof. Let Δ be a maximal simplex in $\Delta \in C_m = \text{Rips}_{r_m}(V) \setminus K_m$, where $K_m = \text{Rips}_{r_{m-1}}(V) \cup T_{r_m}$. It follows from Lemma 24 that the set W_Δ is a discrete gradient on the full subcomplex $\text{Cl } \Delta \subseteq \text{Rips}_{r_m}(V)$ that partitions $\text{St } E_\Delta = \text{Cl } \Delta \setminus (K_m \cap \text{Cl } \Delta)$ and that induces a collapse $\text{Cl } \Delta \searrow (K_m \cap \text{Cl } \Delta)$.

It follows directly from Lemma 21 and Proposition 7 that the union $W_m = \bigcup_\Delta W_\Delta$ is a discrete gradient on $\text{Rips}_{r_m}(V)$. Again by Proposition 7, the union $W = \bigcup_m W_m$ is a discrete gradient on $\text{Cl}(V)$.

By construction of the W_Δ , the union W_m partitions the complement $\text{Rips}_{r_m}(V) \setminus K_m$. Hence, by Proposition 8, it induces a collapse $\text{Rips}_{r_m}(V) \searrow K_m = \text{Rips}_{r_{m-1}}(V) \cup T_{r_m}$. Since only the vertices and the tree edges are critical for W , this also yields the collapse to T_{r_m} . ◀

4.2.2 The perturbed gradient

Assume that V is totally ordered. We construct a coarsening of the canonical gradient to the *perturbed gradient*, such that under a specific total order of V the perturbed gradient is refined by the zero persistence apparent pairs of the diam-lexicographic order $<$ on simplices.

Consider a maximal simplex $\Delta \in C_m$, where $m \in \{1, \dots, l\}$. Note that all edges in E_Δ have length r_m and thus are ordered lexicographically. Enumerate them as $e_1 < \dots < e_q$. Every simplex $\sigma \in C_m \cap \text{Cl } \Delta$ contains a maximal edge $e_\sigma \in \text{Cl } \sigma \cap E_\Delta$.

► **Lemma 26.** *For every edge $e_i \in E_\Delta$ the union $\Sigma_i = \bigcup_{e_\sigma=e_i} \sigma \subseteq \Delta$ is a simplex in C_m and the maximal edge among $\text{Cl } \Sigma_i \cap E_\Delta$ is e_i .*

Proof. Note that $\Sigma_i \subseteq \Delta \in \text{Rips}_{r_m}(V)$ is a simplex and it is contained in C_m , because it is a coface of the non-tree edge e_i of length r_m .

To prove the second claim, let $e_j \in \text{Cl } \Sigma_i \cap E_\Delta$ be any edge. Write $e_i = \{x, y\}$ with $x < y$ and $e_j = \{a, b\}$ with $a < b$. By construction of Σ_i , there exist simplices $\sigma_a, \sigma_b \in C_m \cap \text{Cl } \Delta$ with $a \in \sigma_a, b \in \sigma_b$ and $e_{\sigma_a} = e_{\sigma_b} = e_i$. Note that $\{x, y, a\} \subseteq \sigma_a$ and $\{x, y, b\} \subseteq \sigma_b$.

By Lemma 16, we have $x, y \in S_{r_m}(a) \cup S_{r_m}(b)$ and therefore $d(a, y) = r_m$ (implying $a \neq y$) or $d(b, y) = r_m$ (implying $b \neq y$). As $\{a, y\} \subseteq \sigma_a$ and $\{b, y\} \subseteq \sigma_b$, this implies $\{a, y\} \leq e_{\sigma_a} = e_i = \{x, y\}$ or $\{b, y\} \leq e_{\sigma_b} = e_i = \{x, y\}$, respectively. In particular, we have $a \leq x$ or $a < b \leq x$, and if $a = x$, then $e_j \subseteq \sigma_b$. In any case, $e_j < e_i = e_{\sigma_b}$ as claimed. ◀

This lemma implies that $N_\Delta = \{[e_i, \Sigma_i]\}_{i=1}^q$ is a collection of disjoint intervals. It follows from Lemma 24 that for each $j \in \{1, \dots, q\}$ the interval $[e_j, \Sigma_j]$ is the union

$$[e_j, \Sigma_j] = \bigcup \{[\text{US}, (\text{US}) \cup L_\Delta] \mid S \subseteq E_\Delta, e_j \text{ maximal element of } \text{Cl}(\text{US}) \cap E_\Delta\} \quad (5)$$

and that N_Δ partitions $C_m \cap \text{Cl } \Delta$. Moreover, it is the discrete gradient of the function

$$f_\Delta: \text{Cl } \Delta \rightarrow \mathbb{R}, \sigma \mapsto \begin{cases} i & \sigma \in [e_i, \Sigma_i] \\ \dim \sigma - \dim \Delta & \sigma \in K_m \end{cases} \quad (6)$$

and the intervals are regular, because L_Δ is nonempty (Lemma 22). By Proposition 8, N_Δ induces a collapse $\text{Cl } \Delta \searrow K_m \cap \text{Cl } \Delta$. Therefore, the total order on V induces a symbolic perturbation scheme on the edges, establishing the situation of a generic tree metric as in Section 4.1.

15:12 Gromov Hyperbolicity, Geodesic Defect, and Apparent Pairs in Rips Filtrations

Consider the union $N_m = \bigcup_{\Delta} N_{\Delta}$, where Δ runs over all maximal simplices in C_m . We call $N = \bigcup_m N_m$ the *perturbed gradient*. By (5), the perturbed gradient N coarsens the canonical gradient W . Analogously to Theorem 25, we obtain the following result.

► **Theorem 27.** *The perturbed gradient is a discrete gradient on $\text{Cl}(V)$. For every $m \in \{1, \dots, l\}$, it induces a sequence of collapses*

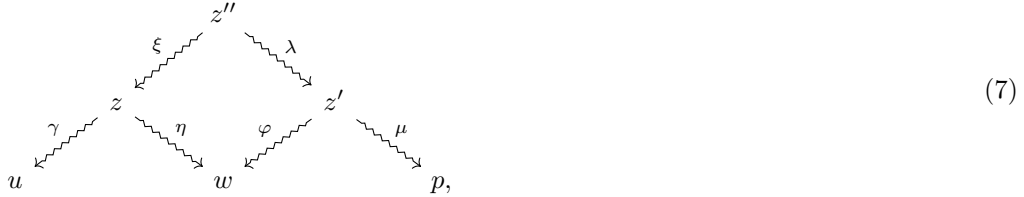
$$\text{Rips}_{r_m}(V) \searrow \text{Rips}_{r_{m-1}}(V) \cup T_{r_m} \searrow T_{r_m}.$$

► **Remark 28.** As the lower bounds of the intervals in the perturbed gradient are edges, it follows from Theorem 27 that these collapses can be expressed as *edge collapses* [9], a notion that is similar to the elementary strong collapses described in Remark 15.

4.2.3 The apparent pairs gradient

Finally, we show that for a specific total order of V , which we describe next, the perturbed gradient is refined by the *zero persistence apparent pairs* of the diam-lexicographic order.

From now on, assume that the tree T is rooted at an arbitrary vertex and orient every edge away from this point. Let \leq_V be the partial order on V where u is smaller than w if there exists an oriented path $u \rightsquigarrow w$. In particular, we have the identity path $\text{id}: u \rightsquigarrow u$. Note that for any two vertices $u, w \in V$ the unique shortest unoriented path $u \rightsquigarrow w$ can be written uniquely as a zig-zag $u \xrightarrow{\gamma} z \xrightarrow{\eta} w$, where z is the greatest point with $z \leq_V u, z \leq_V w$, and γ, η are oriented paths in T that intersect only in z . If $w \rightsquigarrow p$ is another unique shortest unoriented path with the zig-zag $w \xrightarrow{\varphi} z' \xrightarrow{\mu} p$, then we can form the following diagram



where z'' is the greatest point with $z'' \leq_V z, z'' \leq_V z'$. Moreover, as T has no cycles, it follows that either ξ or λ is the identity path and $\varphi \circ \lambda = \eta$ or $\eta \circ \xi = \varphi$, respectively.

Extend the partial order \leq_V on V to a total order $<$ and consider the diam-lexicographic order on simplices. As this total order on the simplices extends $<$ under the identification $v \mapsto \{v\}$, we will also denote it by $<$. The following lemma directly implies Theorem 5.

► **Lemma 29.** *The intervals in the perturbed gradient N are refined by apparent pairs with respect to $<$. For every $m \in \{1, \dots, l\}$, the zero persistence apparent pairs induce the collapse*

$$\text{Rips}_{r_m}(V) \searrow \text{Rips}_{r_{m-1}}(V) \cup T_{r_m}.$$

Proof. Consider a maximal simplex $\Delta \in C_m$. Recall that N_{Δ} is the discrete gradient of the function $f_{\Delta}: \text{Cl} \Delta \rightarrow \mathbb{R}$ defined in (6), using the same vertex order as above. By Lemma 9, the zero persistence apparent pairs with respect to the f_{Δ} -lexicographic order $<_{f_{\Delta}}$ are precisely the gradient pairs of the minimal vertex refinement of N_{Δ} .

We next show that each apparent pair $(\sigma, \tau = \sigma \cup \{v\}) \subseteq [e_i, \Sigma_i]$ with respect to $<_{f_{\Delta}}$, where v is the minimal vertex in $\Sigma_i \setminus e_i$, is an apparent pair with respect to $<$. Clearly, these pairs have persistence zero with respect to the diameter function, as they appear in the same interval of the perturbed gradient. As the apparent pairs of $<_{f_{\Delta}}$, taken over all Δ ,

yield a partition of $C_m = \text{Rips}_{r_m}(V) \setminus (\text{Rips}_{r_{m-1}}(V) \cup T_{r_m})$, the same is then true for the apparent pairs of $<$. Thus, by Proposition 8, the apparent pairs gradient induces a collapse $\text{Rips}_{r_m}(V) \searrow \text{Rips}_{r_{m-1}}(V) \cup T_{r_m}$.

First, let $\sigma \cup \{p\} \in C_m$ be a cofacet of σ not equal to τ . We show that we must have $\tau < \sigma \cup \{p\}$, proving that τ is the minimal cofacet of σ with respect to $<$: If $p \in \Sigma_i$, then $p \in \Sigma_i \setminus e_i$, as $p \notin \sigma \supseteq e_i$, and the statement is true by minimality of v in the minimal vertex refinement. Now assume that $p \notin \Sigma_i$ and write $e_i = \{u, w\}$ with $u < w$. By (5), we have $L_\Delta \subseteq \Sigma_i$ and hence it follows that $p \notin L_\Delta$ and that the point p is contained in an edge in E_Δ , by definition of L_Δ . It follows from Lemma 16 that p together with at least one vertex of e_i forms an edge in E_Δ . Call this edge g ; if there are two such edges, consider the larger one, and call it g . From $\{u, w, p\} \subseteq \Delta$ and $p \notin \Sigma_i$ we get $e_i < g$: The edge e_i is not the maximal edge of the two simplex $\{u, w, p\}$, since otherwise p would be contained in Σ_i . Hence, one of the two other edges is maximal, and that edge is g by definition. Considering the two possible cases $g = \{u, p\}$ and $g = \{w, p\}$, we must have $u < p$. We will argue that $v < p$ holds, which proves $\tau = \sigma \cup \{v\} < \sigma \cup \{p\}$.

Consider the diagram (7). If $\gamma \neq \text{id}$, then it follows from the fact that $e_i = \{u, w\}$ is not a tree edge that along the unique shortest path $u \rightsquigarrow w$ there exists a vertex x distinct from u and w with $x < u < p$. Then $x \in L_\Delta \subseteq \Sigma_i \setminus e_i$ by Lemma 22, and as v is the minimal element in $\Sigma_i \setminus e_i$, we get $v \leq x < p$.

If $\gamma = \text{id}$, then $u = z$, and it follows from $d(w, p) \leq r_m$ and $p \notin e_i = \{u, w\}$ that we must have $\lambda \neq \text{id}$ and $\xi = \text{id}$: Otherwise $\lambda = \text{id}$ and $u = z$ lies on φ . Therefore, u lies on the unique shortest path from w to p and $d(w, p) = d(w, u) + d(u, p) = r_m + d(u, p) > r_m$, yielding a contradiction. Thus, the unique shortest path $(u = z) \rightsquigarrow p$ decomposes as $u \rightsquigarrow z' \rightsquigarrow p$, where $u \rightsquigarrow z'$ is contained in $u \rightsquigarrow z' \rightsquigarrow w$. Note that $u \neq z'$, because $\lambda \neq \text{id}$. Hence, as e_i is not a tree edge, the immediate successor x of u on the path $u \rightsquigarrow w$ is distinct from u and w with $x \leq z'$. This point satisfies $x \leq z' \leq p$, and it follows from Lemma 22 that we have $x \in L_\Delta \subseteq \Sigma_i \setminus e_i$. Because $p \notin L_\Delta$ we even have $x < p$. Therefore, as v is the minimal vertex in $\Sigma_i \setminus e_i$, it follows that $v \leq x < p$.

It remains to prove that σ is the maximal facet of τ with respect to $<$. We write $e_i = \{u, w\}$ with $u < w$ and $\tau = \{b_0, \dots, b_{\dim \tau}\}$ with $b_0 < \dots < b_{\dim \tau}$. As $e_i \subseteq \tau$, there are indices $k_1 < k_2$ with $u = b_{k_1} < b_{k_2} = w$. If $k_1 > 0$, then $v = b_0$, so σ is of the form $\{b_1, \dots, b_{\dim \tau}\}$ and is the maximal facet of τ with respect to $<$ as claimed. Now assume $k_1 = 0$. If τ contains no edges $e \in E_\Delta$ other than e_i , then the facets $\tau \setminus \{u\}$ and $\tau \setminus \{w\}$ are both contained in $\text{Rips}_{r_{m-1}}(V)$, because they do not contain any edge of length r_m , and the maximal facet of τ is $\tau \setminus \{x\}$ with x the minimal vertex in $\tau \setminus e_i$. By assumption, we have $x = v$ and hence $\tau \setminus \{x\} = \tau \setminus \{v\} = \sigma$. If τ contains other edges $e \neq e_i$ with $e \in E_\Delta$, label them s_1, \dots, s_a . As $e_i \subseteq \tau \subseteq \Sigma_i$, it follows from Lemma 26 that we have $s_b < e_i$ for all b . Because of this and our assumption $k_1 = 0$, i.e., u is the minimal vertex of τ , we have $s_b = \{u, x_b\} < \{u, w\} = e_i$ with $u < x_b < w$. Therefore, the facet $\{b_1, \dots, b_{\dim \tau}\}$ contains no edges in E_Δ and hence it is contained in $\text{Rips}_{r_{m-1}}(V)$. The facet $\{b_0, b_2, \dots, b_{\dim \tau}\}$ of τ contains e_i , hence it is an element of C_m , and so it is maximal among the facets containing b_0 , implying that it is the maximal facet of τ with respect to $<$. Because b_1 is the minimal vertex in $\tau \setminus e_i$ and $v \in \tau \setminus e_i$, it follows from the minimality of $v \in \Sigma_i \setminus e_i$ that we have $b_1 = v$, implying $\{b_0, b_2, \dots, b_{\dim \tau}\} = \sigma$. Therefore, σ is the maximal facet of τ with respect to $<$. ◀

► Remark 30. The preceding Lemma 29 also implies Theorem 3 in the special case of tree metrics: if $u > t \geq 2\nu(V) = \max_{e \in E} l(e)$ are real numbers, then $T_t = T$ is the entire tree, and we obtain collapses $\text{Rips}_u(V) \searrow \text{Rips}_t(V) \searrow T \searrow \{*\}$. If all edges of T have the same length, it turns out that the collapse $T \searrow \{*\}$ is also induced by the apparent pairs gradient for the same order $<$.

References

- 1 Michał Adamaszek. Clique complexes and graph powers. *Israel J. Math.*, 196(1):295–319, 2013. doi:10.1007/s11856-012-0166-1.
- 2 Michał Adamaszek, Henry Adams, Ellen Gasparovic, Maria Gommel, Emilie Purvine, Radmila Sazdanovic, Bei Wang, Yusu Wang, and Lori Ziegelmeier. On homotopy types of Vietoris–Rips complexes of metric gluings. *J. Appl. Comput. Topol.*, 4(3):425–454, 2020. doi:10.1007/s41468-020-00054-y.
- 3 Dominique Attali, André Lieutier, and David Salinas. Vietoris–Rips complexes also provide topologically correct reconstructions of sampled shapes. *Comput. Geom.*, 46(4):448–465, 2013. doi:10.1016/j.comgeo.2012.02.009.
- 4 Dominique Attali, André Lieutier, and David Salinas. When convexity helps collapsing complexes. In *35th International Symposium on Computational Geometry*, volume 129 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Art. No. 11, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019. doi:10.4230/LIPICs.SocG.2019.11.
- 5 Jonathan Ariel Barmak and Elias Gabriel Minian. Strong homotopy types, nerves and collapses. *Discrete Comput. Geom.*, 47(2):301–328, 2012. doi:10.1007/s00454-011-9357-5.
- 6 Ulrich Bauer. Ripser: efficient computation of Vietoris–Rips persistence barcodes. *J. Appl. Comput. Topol.*, 5(3):391–423, 2021. doi:10.1007/s41468-021-00071-5.
- 7 Ulrich Bauer and Herbert Edelsbrunner. The Morse theory of Čech and Delaunay complexes. *Trans. Amer. Math. Soc.*, 369(5):3741–3762, 2017. doi:10.1090/tran/6991.
- 8 Michael Bleher, Lukas Hahn, Juan Angel Patino-Galindo, Mathieu Carriere, Ulrich Bauer, Raul Rabadan, and Andreas Ott. Topology identifies emerging adaptive mutations in SARS-CoV-2. Preprint, 2021. arXiv:2106.07292.
- 9 Jean-Daniel Boissonnat and Siddharth Pritam. Edge collapse and persistence of flag complexes. In *36th International Symposium on Computational Geometry*, volume 164 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Art. No. 19, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020. doi:10.4230/LIPICs.SocG.2020.19.
- 10 M. Bonk and O. Schramm. Embeddings of Gromov hyperbolic spaces. In *Selected works of Oded Schramm. Volume 1, 2*, Sel. Works Probab. Stat., pages 243–284. Springer, New York, 2011. With a correction by Bonk. doi:10.1007/978-1-4419-9675-6_10.
- 11 Joseph Minhow Chan, Gunnar Carlsson, and Raul Rabadan. Topology of viral evolution. *Proceedings of the National Academy of Sciences*, 110(46):18566–18571, 2013. doi:10.1073/pnas.1313480110.
- 12 R. Espínola and M. A. Khamsi. Introduction to hyperconvex spaces. In *Handbook of metric fixed point theory*, pages 391–435. Kluwer Acad. Publ., Dordrecht, 2001. doi:10.1007/978-94-017-1748-9_13.
- 13 Steven N. Evans. *Probability and real trees*, volume 1920 of *Lecture Notes in Mathematics*. Springer, Berlin, 2008. Lectures from the 35th Summer School on Probability Theory held in Saint-Flour, July 6–23, 2005. doi:10.1007/978-3-540-74798-7.
- 14 Robin Forman. Morse theory for cell complexes. *Adv. Math.*, 134(1):90–145, 1998. doi:10.1006/aima.1997.1650.
- 15 Ragnar Freij. Equivariant discrete Morse theory. *Discrete Math.*, 309(12):3821–3829, 2009. doi:10.1016/j.disc.2008.10.029.
- 16 M. Gromov. Hyperbolic groups. In *Essays in group theory*, volume 8 of *Math. Sci. Res. Inst. Publ.*, pages 75–263. Springer, New York, 1987. doi:10.1007/978-1-4613-9586-7_3.
- 17 Allen Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, 2002.
- 18 Patricia Hersh. On optimizing discrete Morse functions. *Adv. in Appl. Math.*, 35(3):294–322, 2005. doi:10.1016/j.aam.2005.04.001.
- 19 J. R. Isbell. Six theorems about injective metric spaces. *Comment. Math. Helv.*, 39:65–76, 1964. doi:10.1007/BF02566944.
- 20 Jakob Jonsson. *Simplicial complexes of graphs*, volume 1928 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2008. doi:10.1007/978-3-540-75859-4.

- 21 Dmitry N. Kozlov. *Organized collapse: an introduction to discrete Morse theory*, volume 207 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2020.
- 22 Urs Lang. Injective hulls of certain discrete metric spaces and groups. *J. Topol. Anal.*, 5(3):297–331, 2013. doi:10.1142/S1793525313500118.
- 23 Janko Latschev. Vietoris-Rips complexes of metric spaces near a closed Riemannian manifold. *Arch. Math. (Basel)*, 77(6):522–528, 2001. doi:10.1007/PL00000526.
- 24 Michael Lesnick, Raúl Rabadán, and Daniel I. S. Rosenbloom. Quantifying genetic innovation: mathematical foundations for the topological study of reticulate evolution. *SIAM J. Appl. Algebra Geom.*, 4(1):141–184, 2020. doi:10.1137/18M118150X.
- 25 Sunhyuk Lim, Facundo Memoli, and Osman Berat Okutan. Vietoris-Rips Persistent Homology, Injective Metric Spaces, and The Filling Radius. Preprint, 2020. arXiv:2001.07588.
- 26 Vin de Silva and Gunnar Carlsson. Topological estimation using witness complexes. In Markus Gross, Hanspeter Pfister, Marc Alexa, and Szymon Rusinkiewicz, editors, *SPBG'04 Symposium on Point - Based Graphics 2004*. The Eurographics Association, 2004. doi:10.2312/SPBG/SPBG04/157-166.
- 27 Leopold Vietoris. Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen. *Math. Ann.*, 97(1):454–472, 1927. doi:10.1007/BF01447877.

Acute Tours in the Plane

Ahmad Biniáz   

School of Computer Science, University of Windsor, Canada

Abstract

We confirm the following conjecture of Fekete and Woeginger from 1997: for any sufficiently large even number n , every set of n points in the plane can be connected by a spanning tour (Hamiltonian cycle) consisting of straight-line edges such that the angle between any two consecutive edges is at most $\pi/2$. Our proof is constructive and suggests a simple $O(n \log n)$ -time algorithm for finding such a tour. The previous best-known upper bound on the angle is $2\pi/3$, and it is due to Dumitrescu, Pach and Tóth (2009).

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases planar points, acute tour, Hamiltonian cycle, equitable partition

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.16

Funding Supported by NSERC.

Acknowledgements I am very grateful to the anonymous reviewer who meticulously verified our proof, and provided valuable feedback that reduced the number of subcases to two (which was three in our original proof) and improved the bound on n to 20 (which was 36 originally).

1 Introduction

The Euclidean traveling salesperson problem (TSP) is a well-studied and fundamental problem in combinatorial optimization and computational geometry. In this problem we are given a set of points in the plane and our goal is to find a shortest tour that visits all points. Motivated by applications in robotics and motion planning, in recent years there has been an increased interest in the study of tours with bounded angles at vertices, rather than bounded length of edges; see e.g. [2, 3, 13, 14, 15] and references therein. Bounded-angle structures (tours, paths, trees) are also desirable in the context of designing networks with directional antennas [6, 7, 11, 19]. Bounded-angle tours (and paths), in particular, have received considerable attention following the PhD thesis of S. Fekete [14] and the seminal work of Fekete and Woeginger [15].

Consider a set P of at least three points in the plane. A *spanning tour* is a directed Hamiltonian cycle on P that is drawn with straight-line edges. When three consecutive vertices p_i, p_{i+1}, p_{i+2} of the tour are traversed in this order, the *rotation angle* at p_{i+1} (denoted by $\angle p_i p_{i+1} p_{i+2}$) is the angle in $[0, \pi]$ that is determined by the segments $p_i p_{i+1}$ and $p_{i+1} p_{i+2}$. If all rotation angles in a tour are at most $\pi/2$ then it is called an *acute tour*.

In 1997, Fekete and Woeginger [15] raised many challenging questions about bounded-angle tours and paths. In particular they conjectured that *for any sufficiently large even number n , every set of n points in the plane admits an acute spanning tour (a tour with rotation angles at most $\pi/2$)*. They stated the conjecture specifically for $n \geq 8$. The point set illustrated in Figure 1(a) (also described in [15]) shows that the upper bound $\pi/2$ is the best achievable. The conjecture does not hold if n is allowed to be an odd number; for example if the n points are on a line then in any spanning tour one of the rotation angles must be π . The conjecture also does not hold if n is allowed to be small. For instance the 4-element point set consisting of the 3 vertices of an equilateral triangle with its center, must have a



© Ahmad Biniáz;

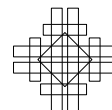
licensed under Creative Commons License CC-BY 4.0

38th International Symposium on Computational Geometry (SoCG 2022).

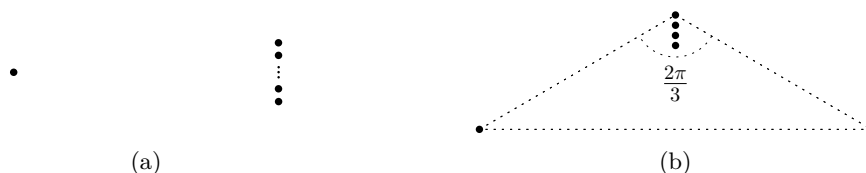
Editors: Xavier Goaoc and Michael Kerber; Article No. 16; pp. 16:1–16:8

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



rotation angle $2\pi/3$ in any spanning tour. Also the 6-element point set of Figure 1(b) (also illustrated in [15] and [13]) must have a rotation angle of at least $2\pi/3 - \epsilon$ in any spanning tour, for some arbitrary small constant ϵ .



■ **Figure 1** (a) a general lower bound example, and (b) a lower bound example for 6 points.

In 2009, Dumitrescu, Pach and Tóth [13] took the first promising steps towards proving the conjecture. They confirmed the conjecture for points in convex position. For general point sets, they obtained the first partial result by showing that any point set (with even number of points) admits a spanning tour in which each rotation angle is at most $2\pi/3$.

In this paper we prove the conjecture of Fekete and Woeginger for general point sets.

► **Theorem 1.** *Let $n \geq 20$ be an even integer. Then every set of n points in the plane admits an acute spanning tour. Such a tour can be computed in linear time after finding an equitable partitioning of points with two orthogonal lines.*

Due to our desire of having a short proof, we prove the conjecture for $n \geq 20$. Perhaps with detailed case analysis one could extend the range of n to a number smaller than 20.

Difficulties towards a proof. Fekete and Woeginger [15] exhibited an arbitrary-large even-size point set for which an algorithm (or a proof technique), that always outputs the longest tour or includes the diameter in the solution, does not achieve an acute tour; the point set is similar to that of Figure 1(b) but has more than 6 points. This somehow breaks the hope for finding an acute tour by using greedy techniques. Therefore, to prove the conjecture one might need to employ some nontrivial ideas.

Related problems

Another interesting conjecture of Fekete and Woeginger [15] is that any set of points in the plane admits a spanning path in which all rotation angles are at least $\pi/6$.¹ In 2008, Bárány, Pór, and Valtr [8] obtained the first constant lower bound of $\pi/9$, thereby gave a partial answer to the conjecture. The full conjecture was then proved, although not yet written in a paper format, by J. Kynčl [16] (see also the note added in the proof of [8]).

Fekete and Woeginger [15] showed that any set of points in the plane admits an acute spanning path (where all intermediate rotation angles are at most $\pi/2$). Such a path can be obtained simply by starting from an arbitrary point and iteratively connecting the current point to its farthest among the remaining points. Notice that the resulting path always contains the diameter and by the difficulties mentioned above it cannot be completed to an acute tour. Carmi et al. [11] showed how to construct acute paths with shorter edges; again no guarantee to be completed to an acute tour. Aichholzer et al. [4] studied a similar problem with an additional constraint that the path should be *plane* (i.e., its edges do not

¹ This bound is the best achievable as the three vertices of an equilateral triangle together with its center do not admit a path with rotation angles greater than $\pi/6$.

cross each other). Among other results, they showed that any set of points in the plane in general position admits a plane spanning path with rotation angles at most $3\pi/4$. They also conjectured that this upper bound could be replaced by $\pi/2$.

The bounded-angle minimum spanning tree (also known as α -MST) is a related problem that asks for a Euclidean minimum spanning tree in which all edges incident to every vertex lie in a cone of angle at most α . This problem is motivated by replacing omni-directional antennas – in a wireless network – with directional antennas which are more secure, require lower transmission ranges, and cause less interference; see e.g. [6, 7, 9, 10, 19].

Another related problem (with an objective somewhat opposite to ours) is to minimize the total *turning angle* of the tour [2].² Similar problems also studied under *pseudo-convex* tours and paths (that make only right turns) [15] and *reflexivity* of a point set (the smallest number of reflex vertices in a simple polygonalization of the point set) [1, 5].

The so-called *Tverberg cycle* is a cycle with straight-line edges such that the diametral disks³ induced by the edges have nonempty intersection. Recently, Pirahmad et al. [17] showed how to construct a spanning Tverberg cycle on any set of points in the plane. Although the constructed cycle has many acute angles, it is still far from being fully acute.

► **Remark.** It is worth mentioning that having a tour with many acute angles, does not necessarily help in getting a fully acute tour because one can simply get a tour with at least $n - 2$ acute angles by interconnecting the endpoints of acute paths obtained in [11, 15].

2 Preliminaries for the proof

A set of four points in the plane is called a *quadruple*. If the four points are in convex position then the quadruple is called *convex*, otherwise it is called *concave*; the quadruple in Figure 2(a) is convex while the quadruples in Figures 2(b) and 2(c) are concave. We refer to the interior point of a concave quadruple as its *center*. By connecting the center of a concave quadruple to its other three points we obtain three angles. If one of these angles is at most $\pi/2$ then the quadruple is called *concave-acute*, otherwise all the angles are larger than $\pi/2$ and the quadruple is called *concave-obtuse*; the quadruple in Figure 2(b) is concave-acute while the one in Figure 2(c) is concave-obtuse.

A path, that is drawn by straight-line edges, is called *acute* if all the angles determined by its adjacent edges are at most $\pi/2$. For two directed paths P_1 and P_2 , where P_1 ends at the same vertex at which P_2 starts, we denote their concatenation by $P_1 \oplus P_2$.

For two distinct points p and q in the plane, we say that p is *to the left of* q if the x -coordinate of p is not larger than the x -coordinate of q . Analogously, we say that p is *below* q if the y -coordinate of p is not larger than the y -coordinate of q .

It is known that any set of n points in the plane can be split into four parts of equal size using two orthogonal lines (see e.g. [18] or [12, Section 6.6]); such lines can be computed in $\Theta(n \log n)$ time [18]. The following is a restatement of this result that is borrowed from [13].

► **Lemma 2.** *Given a set S of n points in the plane (n even), one can always find two orthogonal lines ℓ_1, ℓ_2 and a partition $S = S_1 \cup S_2 \cup S_3 \cup S_4$ with $|S_1| = |S_3| = \lfloor \frac{n}{4} \rfloor$ and $|S_2| = |S_4| = \lceil \frac{n}{4} \rceil$ such that S_1 and S_3 belong to two opposite closed quadrants determined by ℓ_1 and ℓ_2 , and S_2 and S_4 belong to the other two opposite closed quadrants.*

² The turning angle at a vertex v is the change in the direction of motion at v when traveling on the tour. It is essentially π minus the rotation angle at v .

³ The diametral disk induced by an edge pq is the disk that has pq as its diameter.

Our proof of Theorem 1 shares some similarities with that of Dumitrescu et al. [13] (for points in convex position) in the sense that both proofs employ the equitable partitioning of Lemma 2. However, there are major differences between the two proofs mainly because simple structures, that appear in points in convex position, do not necessarily appear in general point sets. Therefore one needs to extract complex structures from general point sets and combine them to establish a proof.

3 Proof of Theorem 1

Throughout this section we assume that n is an even integer. We show how to construct an acute tour on any set of $n \geq 20$ points in the plane, and thus proving Theorem 1. In Subsection 3.1 we describe the setup for our construction, and then in Subsection 3.2 we construct the tour.

3.1 The proof setup

Let S be a set of $n \geq 20$ points in the plane. Let $\{S_1, S_2, S_3, S_4\}$ be an equitable partitioning of S with two orthogonal lines ℓ_1 and ℓ_2 that satisfies the conditions of Lemma 2. After a suitable rotation and translation we may assume that ℓ_1 and ℓ_2 coincide with the x and y coordinate axes, respectively. Also, after a suitable relabeling we may assume that all points of S_i belong to the i th quadrant determined by the axes as depicted in Figure 2(a).

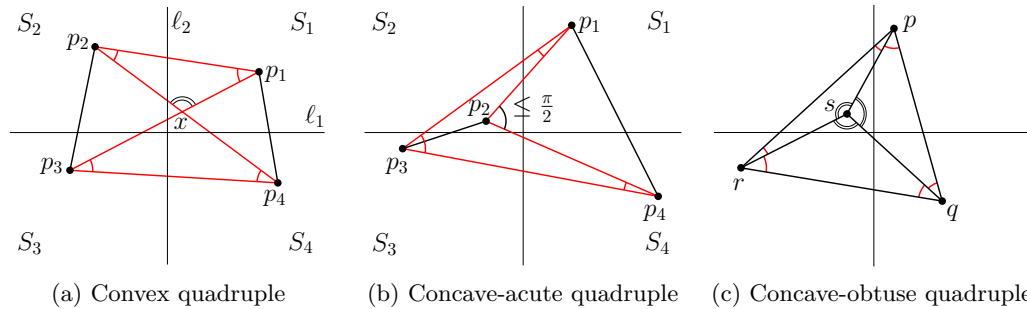


Figure 2 Illustration of (a) Lemma 3 where P is convex and $\angle p_1xp_2 \geq \pi/2$, (b) Lemma 3 where P is concave-acute and $\angle p_1p_2p_4 \leq \pi/2$, and (c) Lemma 4 where all the three angles at s are obtuse.

Based on the above partitioning we introduce four types of quadruples. Let $P = \{p_1, p_2, p_3, p_4\}$ be a quadruple such that $p_i \in S_i$ for all $i = 1, 2, 3, 4$. We say that P is *upward* if the path $p_2p_4p_3p_1$ (or equivalently $p_1p_3p_4p_2$) is acute, *downward* if the path $p_3p_1p_2p_4$ (or equivalently $p_4p_2p_1p_3$) is acute, *leftward* if the path $p_2p_4p_1p_3$ (or equivalently $p_3p_1p_4p_2$) is acute, and *rightward* if the path $p_1p_3p_2p_4$ (or equivalently $p_4p_2p_3p_1$) is acute. Such paths are referred to as “hooks” in [13]. The following lemmas and observation, although very simple, play important roles in our proof.

► **Lemma 3.** *Let $P = \{p_1, p_2, p_3, p_4\}$ be a quadruple such that $p_i \in S_i$ for all $i = 1, 2, 3, 4$. If P is convex or concave-acute then it is upward and downward or it is leftward and rightward.*

Proof. First assume that P is convex. Let x denote the intersection point of the diagonals p_1p_3 and p_2p_4 . If $\angle p_1xp_2 \geq \pi/2$ then the paths $p_2p_4p_3p_1$ and $p_3p_1p_2p_4$ are acute and thus P is upward and downward; see Figure 2(a). If $\angle p_1xp_2 < \pi/2$ then the paths $p_2p_4p_1p_3$ and $p_1p_3p_2p_4$ are acute and thus P is leftward and rightward.

Now assume that P is concave-acute. Without loss of generality we assume that p_2 is the center of P . Observe that in this case $\angle p_1 p_2 p_3$ is obtuse. This and the fact that P is concave-acute imply that one of $\angle p_1 p_2 p_4$ and $\angle p_3 p_2 p_4$ is acute. If $\angle p_1 p_2 p_4$ is acute as depicted in Figure 2(b) then the paths $p_2 p_4 p_3 p_1$ and $p_3 p_1 p_2 p_4$ are acute and thus P is upward and downward (observe that $\angle p_2 p_1 p_3 + \angle p_1 p_3 p_4 + \angle p_3 p_4 p_2 = \angle p_1 p_2 p_4 \leq \pi/2$). Analogously, if $\angle p_3 p_2 p_4$ is acute then the paths $p_2 p_4 p_1 p_3$ and $p_1 p_3 p_2 p_4$ are acute and thus P is leftward and rightward. ◀

► **Lemma 4.** *Let $\{p, q, r, s\}$ be a concave-obtuse quadruple with center s . Then all angles $\angle pqs, \angle qps, \angle qrs, \angle rqs, \angle rps,$ and $\angle prs$ are acute.*

Proof. See Figure 2(c). In each of the triangles $\triangle spq, \triangle sqr,$ and $\triangle srp$ the angle at s is larger than $\pi/2$. Thus the other two angles are acute. ◀

► **Lemma 5.** *Let $P = \{p_1, p_2, p_3, p_4\}$ be a quadruple such that $p_i \in S_i$ for all $i = 1, 2, 3, 4$. If P is concave-obtuse then it is upward, downward, leftward, or rightward.*

Proof. Without loss of generality assume that p_2 is the center of P . See Figure 2(c) where $p_2 = s$. In the triangle $\triangle p_1 p_3 p_4$ the angle at p_1 or the angle at p_3 is acute. If the angle at p_1 is acute then the path $p_2 p_4 p_1 p_3$ is acute and thus P is leftward ($\angle p_2 p_4 p_1$ is acute by Lemma 4). If the angle at p_3 is acute then the path $p_2 p_4 p_3 p_1$ is acute and thus P is upward ($\angle p_2 p_4 p_3$ is acute by Lemma 4). ◀

► **Observation 6.** *Let $p, q,$ and r be any three points in S such that q and r lie in the quadrant that is opposite to the quadrant containing p . Then the angle $\angle qpr$ is acute.*

3.2 The tour construction

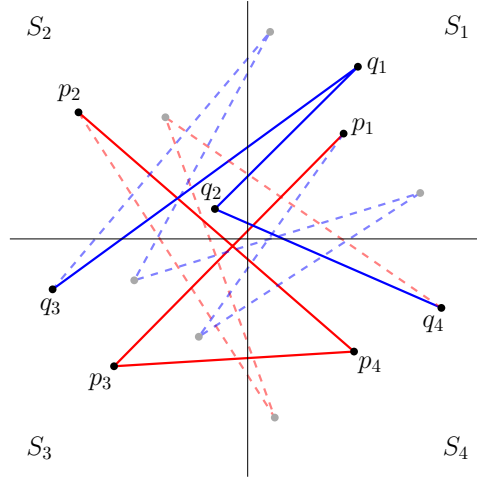
In this section we show how to construct an acute tour on S where $|S| \geq 20$. By Lemma 2 each S_i with $i \in \{1, 2, 3, 4\}$ has at least $\lfloor 20/4 \rfloor = 5$ points. From each S_i we select an arbitrary subset of 5 points, and then we partition (the total 20) selected points into 5 quadruples such that each quadruple contains exactly one point from each S_i . Let \mathcal{Q} denote the set of these quadruples. For any quadruple X in \mathcal{Q} we denote the points of X by x_1, x_2, x_3, x_4 where $x_i \in S_i$ for all $i = 1, 2, 3, 4$.

Since $|\mathcal{Q}| \geq 5$, by the pigeonhole principle \mathcal{Q} has three quadruples that are *vertical* (i.e. upward, downward, or both upward and downward) or three that are *horizontal* (i.e. leftward, rightward, or both leftward and rightward). Without loss of generality assume that \mathcal{Q} has three vertical quadruples. If two of these vertical quadruples are of opposite types, i.e. one upward and one downward, then we construct a tour as in case 1 below. Otherwise, the three quadruples are concave-obtuse and of the same type in which case we construct a tour as in case 2 below. Our constructions take linear time in both cases.

Case 1: \mathcal{Q} contains two quadruples such that one is upward and the other is downward. Let P and Q be such quadruples where P is upward and Q is downward. Since P is upward, the path $p_1 p_3 p_4 p_2$ is acute. Since Q is downward, the path $q_4 q_2 q_1 q_3$ is acute; see Figure 3. Let $\overline{S_2 S_4}$ be a polygonal path starting from p_2 , ending in q_4 , alternating between S_2 and S_4 , and containing all points of $S_2 \cup S_4$ except for q_2 and p_4 . Let $\overline{S_3 S_1}$ be a polygonal path starting from q_3 , ending in p_1 , alternating between S_3 and S_1 , and containing all points of $S_3 \cup S_1$ except for p_3 and q_1 . Such polygonal paths exist because by Lemma 2 we have

16:6 Acute Tours in the Plane

$|S_2| = |S_4|$ and $|S_1| = |S_3|$. All intermediate angles of these two polygonal paths are acute by Observation 6. Then the tour $p_1p_3p_4p_2 \oplus \overline{S_2S_4} \oplus q_4q_2q_1q_3 \oplus \overline{S_3S_1}$ is acute, and it spans S . Notice that the angles at p_1, p_2, q_3 and q_4 are acute by Observation 6.



■ Figure 3 Illustration of Case 1.

Case 2: Q contains three concave-obtuse quadruples of the same type. Let P, Q and R be such quadruples, and without loss of generality assume that they are upward. Thus, the paths $p_2p_4p_3p_1$ and $q_2q_4q_3q_1$ and $r_2r_4r_3r_1$ are acute. Since P, Q and R are concave-obtuse their centers should lie at endpoints of these paths (the centers cannot be interior vertices of acute paths). Thus the center of P is either p_1 or p_2 , the center of Q is either q_1 or q_2 , and the center of R is either r_1 or r_2 . This means that the centers lie in quadrants 1 and 2. By the pigeonhole principle, and after a suitable reflection, we may assume that at least two of the centers lie in quadrant 2. After a suitable relabeling assume that the centers of P and Q (i.e. p_2 and q_2) lie in quadrant 2. The center of R lies either in quadrant 2 (i.e. it is r_2) or in quadrant 1 (i.e. it is r_1).

After a suitable relabeling assume that p_2 lies below q_2 , as in Figure 4. Now we build our tour as follows. First we connect p_2 to p_1 and q_1 . The point p_2 is below p_1 because p_2 lies below the segment p_1p_3 . The point p_2 is also below q_1 because p_2 is below q_2 which is in turn below q_1 (as q_2 lies below the segment q_1q_3). Thus p_2 is below both p_1 and q_1 . Also notice that p_2 is to the left of both p_1 and q_1 . Thus, the angle $\angle p_1p_2q_1$ is acute (imagine moving the origin to p_2 , then both p_1 and q_1 would lie in the first quadrant). Then we connect q_3 to q_1 and q_4 . The angle $\angle q_4q_3q_1$ is acute because Q is upward (i.e. the path $q_2q_4q_3q_1$ is acute). The angle $\angle p_2q_1q_3$ is acute because both p_2 and q_3 lie below and to the left of q_1 . Therefore, the path $p_1p_2q_1q_3q_4$ is acute; see Figure 4. In the rest of the construction we distinguish two subcases, depending on the center of R .

Subcase 2.1: *The center of R is r_1 .* This case is depicted in Figure 4(a). We connect r_4 to r_2 and r_3 . The resulting path $r_2r_4r_3$ is acute (because R is upward, i.e. the path $r_2r_4r_3r_1$ is acute). Let $\overline{S_4S_2}$ be a polygonal path starting from q_4 , ending in r_2 , alternating between S_4 and S_2 , containing all points of $S_4 \cup S_2$ except for r_4, p_2 , and having q_4q_2 as its first edge. Let $\overline{S_3S_1}$ be a polygonal path starting from r_3 , ending in p_1 , alternating between S_3 and S_1 , containing all points of $S_3 \cup S_1$ except for q_3, q_1 , and having r_3r_1 as its first edge and p_3p_1

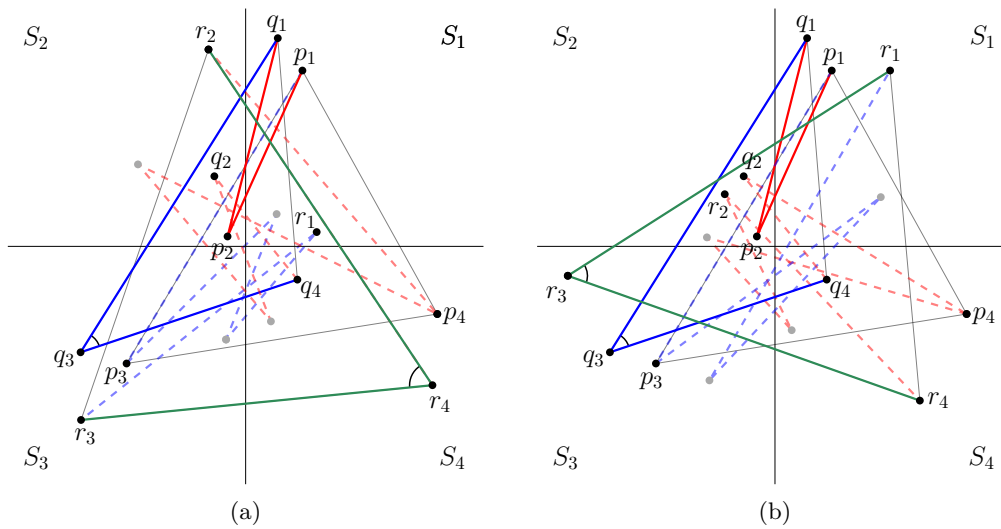


Figure 4 Illustration of Case 2. Three concave-obtuse quadruples P , Q and R that are upward, and the centers of P and Q lie in quadrant 2. (a) Subcase 2.1 where the center of R is in quadrant 1. (b) Subcase 2.2 where the center of R is in quadrant 2.

as its last edge. All intermediate angles of these two paths are acute by Observation 6. By interconnecting the constructed paths we obtain the tour $p_1p_2q_1q_3q_4 \oplus \overline{S_4S_2} \oplus r_2r_4r_3 \oplus \overline{S_3S_1}$ which is acute, and it spans S . The angles at p_1, r_3, q_4 are acute by Lemma 4, and the angle at r_2 is acute by Observation 6.

Subcase 2.2: *The center of R is r_2 .* This case is depicted in Figure 4(b). We connect r_3 to r_4 and r_1 . The resulting path $r_4r_3r_1$ is acute (because R is upward, i.e. the path $r_2r_4r_3r_1$ is acute). Let $\overline{S_4S_2S_4}$ be a polygonal path starting from q_4 , ending in r_4 , alternating between S_4 and S_2 , containing all points of $S_4 \cup S_2$ except for p_2 , and having q_4q_2 as its first edge and r_2r_4 as its last edge. Let $\overline{S_1S_3S_1}$ be a polygonal path starting from r_1 , ending in p_1 , alternating between S_1 and S_3 , containing all points of $S_1 \cup S_3$ except for q_1, q_3, r_3 , and having p_3p_1 as its last edge. Intermediate angles of these paths are acute by Observation 6. Thus $p_1p_2q_1q_3q_4 \oplus \overline{S_4S_2S_4} \oplus r_4r_3r_1 \oplus \overline{S_1S_3S_1}$ is an acute spanning tour. The angles at q_4, r_4 , and p_1 are acute by Lemma 4, and the angle at r_1 is acute by Observation 6. This finishes our proof of Theorem 1.

4 Concluding remarks

We showed how to construct an acute tour on any set of n points in the plane, where n is even and at least 20. Our construction uses at most 12 points in each case (namely the points of quadruples P, Q and R). One might be interested to extend the range of n (to smaller even numbers) by taking advantage of the 8 unused points, although this may require some case analysis.

References

- 1 Eyal Ackerman, Oswin Aichholzer, and Balázs Keszegh. Improved upper bounds on the reflexivity of point sets. *Computational Geometry: Theory and Applications*, 42(3):241–249, 2009.

- 2 Alok Aggarwal, Don Coppersmith, Sanjeev Khanna, Rajeev Motwani, and Baruch Schieber. The angular-metric traveling salesman problem. *SIAM Journal on Computing*, 29(3):697–711, 1999. Also in *SODA'97*.
- 3 Oswin Aichholzer, Anja Fischer, Frank Fischer, J. Fabian Meier, Ulrich Pferschy, Alexander Pilz, and Rostislav Staněk. Minimization and maximization versions of the quadratic travelling salesman problem. *Optimization*, 66(4):521–546, 2017.
- 4 Oswin Aichholzer, Thomas Hackl, Michael Hoffmann, Clemens Huemer, Attila Pór, Francisco Santos, Bettina Speckmann, and Birgit Vogtenhuber. Maximizing maximal angles for plane straight-line graphs. *Computational Geometry: Theory and Applications*, 46(1):17–28, 2013.
- 5 Esther M. Arkin, Sándor P. Fekete, Ferran Hurtado, Joseph S. B. Mitchell, Marc Noy, Vera Sacristán, and Saurabh Sethia. On the reflexivity of point sets. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry: The Goodman-Pollack Festschrift*, pages 139–156. Springer, 2003.
- 6 Rom Aschner and Matthew J. Katz. Bounded-angle spanning tree: Modeling networks with angular constraints. *Algorithmica*, 77(2):349–373, 2017. Also in *ICALP'14*.
- 7 Rom Aschner, Matthew J. Katz, and Gila Morgenstern. Do directional antennas facilitate in reducing interferences? In *Proceedings of the 13th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, pages 201–212, 2012.
- 8 Imre Bárány, Attila Pór, and Pavel Valtr. Paths with no small angles. *SIAM Journal on Discrete Mathematics*, 23(4):1655–1666, 2009. Also in *LATIN'08*.
- 9 Ahmad Biniiaz, Prosenjit Bose, Anna Lubiw, and Anil Maheshwari. Bounded-angle minimum spanning trees. *Algorithmica*, 84(1):150–175, 2022. Also in *SWAT'20*.
- 10 Ahmad Biniiaz, Majid Daliri, and Amir Hossein Moradpour. A 10-approximation of the $\frac{\pi}{2}$ -MST. In *39th International Symposium on Theoretical Aspects of Computer Science (STACS)*, 2022.
- 11 Paz Carmi, Matthew J. Katz, Zvi Lotker, and Adi Rosén. Connectivity guarantees for wireless networks with directional antennas. *Computational Geometry: Theory and Applications*, 44(9):477–485, 2011.
- 12 R. Courant and H. Robbins. *What is Mathematics? An Elementary Approach to Ideas and Methods*. Oxford University Press, New York, 1979.
- 13 Adrian Dumitrescu, János Pach, and Géza Tóth. Drawing Hamiltonian cycles with no large angles. *The Electronic Journal of Combinatorics*, 19(2):P31, 2012. Also in *GD'09*.
- 14 Sándor P. Fekete. *Geometry and the Traveling Salesman Problem*. Phd thesis, University of Waterloo, 1992.
- 15 Sándor P. Fekete and Gerhard J. Woeginger. Angle-restricted tours in the plane. *Computational Geometry: Theory and Applications*, 8:195–218, 1997.
- 16 Jan Kynčl. Personal communication, 2019.
- 17 Olimjoni Pirahmad, Alexandr Polyanskii, and Alexey Vasilevskii. On a Tverberg graph. *CoRR*, abs/2108.09795, 2021. [arXiv:2108.09795](https://arxiv.org/abs/2108.09795).
- 18 Sambuddha Roy and William Steiger. Some combinatorial and algorithmic applications of the borsuk-ulam theorem. *Graphs and Combinatorics*, 23(Supplement-1):331–341, 2007.
- 19 Tien Tran, Min Kyung An, and Dung T. Huynh. Antenna orientation and range assignment algorithms in directional WSNs. *IEEE/ACM Transaction on Networking*, 25(6):3368–3381, 2017. Also in *INFOCOM'16*.

ETH-Tight Algorithms for Finding Surfaces in Simplicial Complexes of Bounded Treewidth

Mitchell Black ✉


School of Electrical Engineering and Computer Science,
Oregon State University, Corvallis, OR, USA

Nello Blaser ✉ 

Department of Informatics, University of Bergen, Norway

Amir Nayyeri ✉

School of Electrical Engineering and Computer Science,
Oregon State University, Corvallis, OR, USA

Erlend Raa Vågset ✉ 

Department of Informatics, University of Bergen, Norway

Abstract

Given a simplicial complex with n simplices, we consider the CONNECTED SUBSURFACE RECOGNITION (c-SR) problem of finding a subcomplex that is homeomorphic to a given connected surface with a fixed boundary. We also study the related SUM-OF-GENUS SUBSURFACE RECOGNITION (SoG) problem, where we instead search for a surface whose boundary, number of connected components, and total genus are given. For both of these problems, we give parameterized algorithms with respect to the treewidth k of the Hasse diagram that run in $2^{O(k \log k)} n^{O(1)}$ time. For the SoG problem, we also prove that our algorithm is optimal assuming the exponential-time hypothesis. In fact, we prove the stronger result that our algorithm is ETH-tight even without restriction on the total genus.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Algebraic topology; Theory of computation → Design and analysis of algorithms

Keywords and phrases Computational Geometry, Surface Recognition, Treewidth, Hasse Diagram, Simplicial Complexes, Low-Dimensional Topology, Parameterized Complexity, Computational Complexity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.17

Related Version *Full Version:* <https://arxiv.org/abs/2203.07566> [5]

Funding *Mitchell Black:* This author was supported in part by NSF grants CCF-1941086 and CCF-1816442.

Amir Nayyeri: This author was supported in part by NSF grants CCF-1941086 and CCF-1816442.

Erlend Raa Vågset: This author was supported in part by the Research Council of Norway grant “Parameterized Complexity for Practical Computing (PCPC)” (NFR, no. 274526).

1 Introduction

Simplicial complexes are a generalization of graphs that give a discrete representation of higher-dimensional spaces. A natural and interesting class of such spaces are manifolds. A d -manifold is a space that is “locally d -dimensional”, meaning each point has a neighborhood homeomorphic to \mathbb{R}^d . Circles and spheres are prototypical examples of 1- and 2-manifolds respectively. Manifolds are important in both mathematics and computer science. For example, triangular meshes in computer graphics are typically 2-manifolds, and the manifold hypothesis in machine learning is the assumption that real-world data often lie on low-dimensional submanifolds of high-dimensional spaces.



© Mitchell Black, Nello Blaser, Amir Nayyeri, and Erlend Raa Vågset;
licensed under Creative Commons License CC-BY 4.0

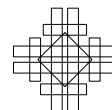
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 17; pp. 17:1–17:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

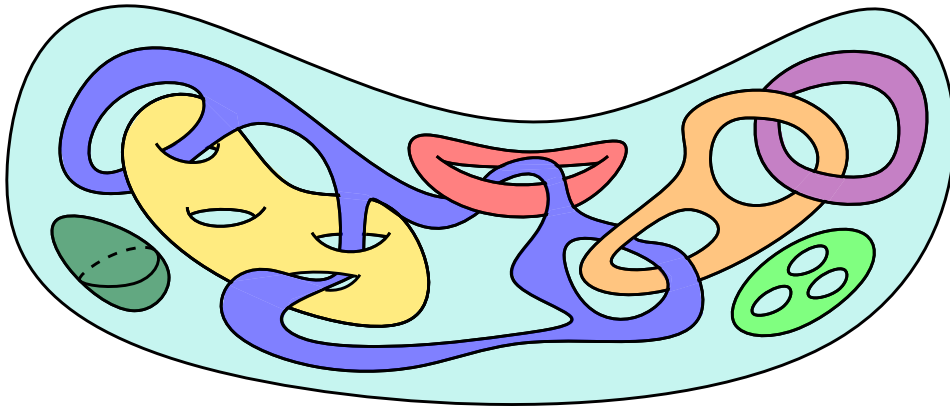


Since manifolds are so important, it is natural to ask if a given simplicial complex is a manifold, or whether two manifolds are homeomorphic. There are fascinating complexity results on these problems. While both recognizing and classifying a 2-manifold have polynomial algorithms, this problem becomes much harder for arbitrary d -manifolds. Deciding whether two manifolds are homeomorphic is undecidable for $d \geq 4$ [20]. Deciding whether or not a simplicial complex is homeomorphic to the d -sphere is undecidable for $d \geq 5$ (see [13]), which implies deciding whether or not a simplicial complex is an n -manifold is undecidable for $d \geq 6$.

We consider several variants of the problem of finding subcomplexes homeomorphic to 2-manifolds, or *surfaces*, in simplicial complexes. While there are polynomial time algorithms for deciding if a simplicial complex is homeomorphic to a surface or deciding the homeomorphism class of a surface, it is a hard problem deciding whether or not a simplicial complex contains a surface as a subcomplex. In particular, Ivanov proved that it is NP-Hard to decide if a simplicial complex contains a 2-sphere [19], and Burton et al. proved that finding a 2-sphere is W[1]-hard when parameterized by solution size [9]. The complexity of this problem is analogous to the graph isomorphism problem. While there is a quasipolynomial algorithm to determine if two graphs are isomorphic [3], it is NP-Hard to determine if one graph contains a subgraph isomorphic to another graph [14].

As this problem is NP-Hard, it is natural to ask whether there are any class of simplicial complexes for which polynomial time algorithms exist. In this paper, we consider the parameterized complexity of this problem and related problems with respect to the treewidth of the Hasse diagram. A tree decomposition of the Hasse diagram defines a recursive series of subcomplexes of K that we can use to incrementally build our surfaces. We also give tight lower bounds for a subset of our algorithms based on the Exponential Time Hypothesis.

1.1 Subsurface recognition problems



■ **Figure 1** A solution to an instance of the SUBSURFACE RECOGNITION problem where we have found an orientable surface consisting of seven connected components with genus 0, 1, 1, 2, 3, 3 and 4 respectively.

We consider several variants of the following generic problem: given a 2-dimensional simplicial complex K and a 1-dimensional subcomplex $B \subset K$, does K contain a subcomplex homeomorphic to a surface with boundary B ? Note that this includes finding surfaces without boundary, as we can set $B = \emptyset$.

The Subsurface Recognition (SR) problem places the most restrictions on the manifold we are looking for. In this problem, we are asked to find a subcomplex of K homeomorphic to a given (possibly disconnected) surface X . Figure 1 shows an example of SR.

► **Problem 1.** *THE SUBSURFACE RECOGNITION (SR) problem:*

Input: A simplicial complex K , a subcomplex $B \subset K$, and a surface X .

Question: Does K contain a subcomplex homeomorphic to X with boundary B ?

Although there is no known FPT algorithm for SR, several variants of SR with looser requirements admit FPT algorithms. One special case of SR requires the surface X to be connected. This variant is called the CONNECTED SUBSURFACE RECOGNITION (c-SR) problem. The extra requirement of connectivity allows us to find an FPT algorithm.

► **Problem 2.** *The CONNECTED SUBSURFACE RECOGNITION (c-SR) problem:*

Input: A simplicial complex K , a subcomplex $B \subset K$, and a connected surface X .

Question: Does K contain a subcomplex homeomorphic to X with boundary B ?

We can also ask for a surface of a certain genus and orientability in K , which is a slightly weaker criterion than finding a surface up to homeomorphism. For a disconnected surface, we define its **total genus** to be the sum of the genus of each of its connected components¹. While a connected surface is characterized up to homeomorphism by its genus and orientability, this is not true for disconnected surfaces. As an example, consider a surface X that is a genus 2 surface and a surface Y that is the disjoint union of two tori. The two surfaces both have total genus 2, but they are not homeomorphic.

► **Problem 3.** *The SUM-OF-GENUS SUBSURFACE RECOGNITION (SoG) problem:*

Input: A simplicial complex K , a subcomplex $B \subset K$, and integers g and c .

Question: Does K contain a surface X of total genus g with c connected components and with boundary B ?

The SUBSURFACE PACKING problem asks to find *any* set of c disjoint surfaces. In particular, no restriction is placed on the genus or orientability of these surfaces.

► **Problem 4.** *The SUBSURFACE PACKING (SP) problem:*

Input: A simplicial complex K , a subcomplex B , and an integer c .

Question: Does K contain a surface X with c connected components and boundary B ?

1.2 Our results

■ **Table 1** Upper and ETH lower bounds for times to solve the different problems considered in this manuscript. Here n is the number of simplices and k is the treewidth of the Hasse diagram. The results of this paper are highlighted.

Problem	SR	c-SR	SoG	SP
Upper	$2^{O(n)}$	$2^{O(k \log k)} n^{O(1)}$	$2^{O(k \log k)} n^{O(1)}$	$2^{O(k \log k)} n^{O(1)}$
Lower	$2^{\omega(k \log k)} n^{O(1)}$	NP-Hard [19]	$2^{\omega(k \log k)} n^{O(1)}$	$2^{\omega(k \log k)} n^{O(1)}$

We consider the parameterized complexity of the above problems with respect to the treewidth k of the Hasse diagram. Table 1 summarizes the known upper and lower bounds. The results of this paper are highlighted. We give FPT algorithms for c-SR, SoG, and SP, and ETH-based lower bounds for SR, SP, and SoG. In fact, we show that these lower bounds are true even when k is the pathwidth of the Hasse diagram. The algorithms for SoG and SP are ETH-tight.

¹ If any connected component of a surface is non-orientable, we will add twice the genus of any orientable components.

1.3 Related work

Tree decompositions and simplicial complexes

Tree decompositions have seen much success as an algorithmic tool on graphs. Often, graphs having tree decompositions of bounded-width admit polynomial-time solutions to otherwise hard problems. A highlight of the algorithmic application of tree decompositions is Courcelle’s Theorem [15], which states that any problem that can be stated in monadic second order logic can be solved in linear time on graphs with bounded treewidth. We recommend [16, Chapter 7] for an introduction to the algorithmic use of tree decompositions.

While tree decompositions have long been successful for algorithms on graphs, they have only recently seen attention for algorithms on simplicial complexes. Existing algorithms use tree decompositions of a variety of graphs associated with a simplicial complex. The most commonly used graph is the dual graph of combinatorial d -manifolds [4, 10, 11, 12]. Other graphs that have been used are level d of the Hasse diagram [11, 7, 6], the adjacency graph of the d -simplices [7], and the 1-skeleton [4]. Our algorithm uses a tree decomposition of the entire Hasse diagram. As far as we know, we are the first to consider tree decompositions of the full Hasse diagram. The condition on vertex links that makes a simplicial complex a surface is dependent on the incidence of vertices and triangles (see Section 2.2), so considering only one level of the Hasse diagram would likely not be sufficient for our problem.

Normal surface theory

Normal surface theory is the study of which surfaces exist as submanifolds of a given 3-manifold. Many algorithms on 3-manifolds, like those for unknot recognition [18] and 3-sphere recognition [21, 22], use normal surface theory. While normal surface theory appears to be similar to our problems, the distinction is that the surfaces in normal surface theory are not subcomplexes of the 3-manifold and can instead intersect 3-simplices in the manifold. Accordingly, the techniques in normal surface theory are quite different from the algorithms we present in this paper.

2 Background

2.1 Simplicial complexes and directed graphs

A **simplicial complex** is a set K such that (1) each element $\sigma \in K$ is a finite set and (2) for each $\sigma \in K$, if $\tau \subset \sigma$, then $\tau \in K$. An element $\sigma \in K$ is a **simplex**. A simplex σ is a **face** of a simplex τ if $\sigma \subset \tau$. Likewise, τ is a **coface** of σ . The simplices σ and τ are **incident**. Two simplices σ_1 and σ_2 are **adjacent** if they are both the face or coface of a simplex τ .

A simplex σ with $|\sigma| = d + 1$ is a **d -simplex**. The set of all d -simplices in K is denoted K_d . The **dimension** of a simplicial complex is the largest integer d such that K contains a d -simplex. A d -dimensional simplicial complex K is **pure** if each simplex in K is a face of d -simplex. We call a 0-simplex a **vertex**, a 1-simplex an **edge**, and a 2-simplex a **triangle**.

The **Hasse diagram** of K is a graph H with vertex set K and edges between each d -simplex $\sigma \in K$ and each $(d - 1)$ -dimensional face of σ for all $d > 0$.

Let $\Sigma \subset K$. The **closure** of Σ is $\text{cl } \Sigma := \{\tau \subset \sigma \mid \sigma \in \Sigma\}$. Note that the closure of Σ is a simplicial complex, even if Σ is not. Note also that the closure $\text{cl } \Sigma$ is defined only by the set Σ and not the complex K . The **star** of Σ is $\text{st}_K \Sigma := \{\sigma \in K \mid \exists \tau \in \Sigma \text{ such that } \tau \subset \sigma\}$.

The **link** of a simplex σ is $\text{lk}_K \sigma = \text{clst}_K \sigma - \text{st}_K \text{cl} \sigma$. Alternatively, the link $\text{lk}_K \sigma$ is all simplices in $\text{clst}_K \sigma$ that do not intersect σ . Note that for any simplex $\tau \in \text{lk}_K \sigma$ that σ and τ are incident to a common coface in $\text{st}_K \sigma$.

A **simple path** is a 1-dimensional simplicial complex $P = \{\{v_1\}, \{v_1, v_2\}, \{v_2\}, \dots, \{v_l\}\}$ such that the vertices $\{v_i\}$ are distinct. The vertices $\{v_1\}, \{v_l\}$ are the **endpoints** of P . We will denote a simple cycle as a tuple $P = (v_1, \dots, v_l)$ as the edges are implied by the vertices. A **simple cycle** is a simple path, with the exception that the endpoints $v_1 = v_l$. We denote a simple cycle with an overline, e.g. $\overline{(v_1, \dots, v_l)}$.

A directed graph D consists of a set of vertices and a set of directed edges, i.e. ordered pairs of vertices $(u, v) := uv$ so that $uv \neq vu$. A **directed simple cycle** C in D (not to be confused with a simple cycle) is a sequence of directed edges $(v_1 v_2, v_2 v_3, \dots, v_l v_1)$ where all the vertices v_i are all distinct. We say that C has the vertex set $\{v_1, \dots, v_l\}$. Two cycles, C and C' , are said to be **vertex disjoint** if their vertex sets are disjoint. A family of cycles is said to be vertex disjoint if they are pairwise vertex disjoint.



■ **Figure 2** Left: A combinatorial surface. The vertex v is an interior vertex. Right: A vertex v with link that is neither a simple path or cycle. We conclude that S is not a combinatorial surface. The point v has no neighborhood homeomorphic to the plane or half-plane, so S is not “locally 2-dimensional” at v .

2.2 Surfaces

Informally, a **surface with boundary** is a compact topological space where each point has a neighborhood homeomorphic to the plane or the half plane, and the **boundary** of the surface is all points with a neighborhood homeomorphic to the half plane. Intuitively, a surface is “locally 2-dimensional”.

Any connected surface with boundary can be constructed by adding handles, crosscaps, and boundary components to a sphere. A **handle** is constructed by removing two disjoint disks from a surface and identifying the boundaries of the removed disks. A **crosscap** is constructed by taking the disjoint union of the surface and the real projective plane, removing a disk from each, and identifying the boundaries of the removed disks. A **boundary component** is constructed by removing a disk from a surface. A surface is **non-orientable** if it has a crosscap and **orientable** otherwise. The **genus** of an orientable surface is the number of handles on the surface, and the genus of a non-orientable surface is the number of crosscaps plus twice the number of handles.

In this paper, we are only concerned with surfaces that are also simplicial complexes, which we call combinatorial surfaces. A **combinatorial surface with boundary** is a pure 2-dimensional simplicial complex S such that the link of each vertex is a simple path or a simple cycle. The condition on the link of the vertices is the combinatorial way of saying that a combinatorial surface is “locally 2-dimensional”. A vertex $v \in S$ such that $\text{lk}_S v$ is a simple path is a **boundary vertex**. A vertex $v \in S$ such that $\text{lk}_S v$ is a simple cycle is an **interior vertex**. Figure 2 shows examples of an interior vertex and a vertex that is neither

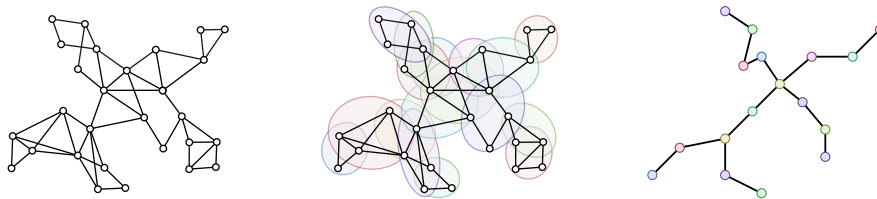
an interior or boundary vertex. It follows from the condition on the links of the vertices that each edge $e \in S$ has link $\text{lk}_S e$ that is either one or two vertices. An edge $e \in S$ such that $\text{lk}_S e$ is a single vertex is a **boundary edge**. An edge $e \in S$ such that $\text{lk}_S e$ is two vertices is an **interior edge**. A triangle $t \in S$ has empty link $\text{lk}_S t = \emptyset$ as S is a 2-dimensional simplicial complex. We denote the set of boundary vertices and boundary edges ∂S . The boundary ∂S is a collection of simple cycles.

2.3 Tree decompositions

Let $G = (V, E)$ be a graph. A **tree decomposition** of G is a tuple (T, X) , where $T = (I, F)$ is a tree with nodes I and edges F , and $X = \{X_t \subset V \mid t \in I\}$ such that (1) $\cup_{t \in I} X_t = V$, (2) for any $\{v_1, v_2\} \in E$, $\{v_1, v_2\} \subset X_t$ for some $t \in I$, and (3) for any $v \in V$, the subtree of T induced by the nodes $\{t \in I \mid v \in X_t\}$ is connected. A set X_t is the **bag** of T . The **width** of (T, X) is $\max_{t \in I} |X_t| - 1$. The **treewidth** of a graph G is the minimum width of any tree decomposition of G . Computing the treewidth of a graph is NP-hard [2], but there are algorithms to compute tree decompositions that are within a constant factor of the treewidth, e.g. [8].

Tree decompositions are used to perform dynamic programs on graphs, and a certain type of tree decomposition, called a nice tree decomposition, makes defining dynamic programs easier. A **nice tree decomposition** is a tree decomposition with a specified root $r \in I$ such that (1) $X_r = \emptyset$, (2) $X_l = \emptyset$ for all leaves $l \in I$, and (3) all non-leaf nodes are either an introduce node, a forget node, or a join node, which are defined as follows. An **introduce node** is a node $t \in I$ with exactly one child t' , and for some $w \in V$, $w \notin X_{t'}$ and $X_t = X_{t'} \cup \{w\}$. We say t **introduces** w . A **forget node** is a node $t \in I$ with exactly one child t' , and for some $w \in V$, $w \in X_t$ and $X_t \setminus \{w\} = X_{t'}$. We say t **forgets** w . A **join node** is a node $t \in I$ with exactly two children t' and t'' where $X_t = X_{t'} = X_{t''}$. The following lemma proves that we can convert any tree decomposition to a nice tree decomposition without increasing width.

► **Lemma 1** (Lemma 7.4 of [16]). *Given a tree decomposition $(T = (I, F), X)$ of width k of a graph $G = (V, E)$, a nice tree decomposition of width k with $O(kn)$ nodes can be computed in $O(k^2 \max\{|V|, |I|\})$ time.*



■ **Figure 3** Left: A graph. Right and Center: A (not nice) tree decomposition of the graph of width 3. Each node of the tree corresponds to a subset of the vertices of the graph.

A **path decomposition** is a special kind of tree decomposition (T, X) where T is a path. A **nice path decomposition** is a tree decomposition without join nodes, i.e. where every node is either an introduce node or a forget node. The **pathwidth** of a graph G is the smallest width of any path decomposition of G . As any path decomposition is also a tree decomposition, the treewidth of G is at most the pathwidth of G .

2.4 The exponential time hypothesis

When a new algorithm is discovered it is natural to ask if it is possible to improve it. To prove that the algorithm was sub-optimal it is enough to find a new and better algorithm. On the other hand, if the algorithm is actually the best possible, then the situation becomes more complicated. Although there are optimality results for a few problems in P ,² none are known for algorithms solving NP-complete problems. Such a result would imply $P \neq NP$, which remains famously unproven.

This theoretical barrier does not make the question of optimality less relevant. No one wants to spend years searching for improvements to an algorithm that cannot be improved! For instance, the algorithms in this paper need $2^{O(k \log k)} n^{O(1)}$ time, which may prompt the question “Why were you unable to deliver a $2^{O(k)} n$ time solution?”.

A pragmatic and popular response to these kinds of questions is to prove that you have optimality under the Exponential Time Hypothesis (ETH). The ETH is a conjecture stating that there is no sub-exponential algorithm for 3-SAT. More precisely, let n be the number of variables in a given instance of 3-SAT.

► **Hypothesis 1 (ETH).** *3-SAT cannot be solved in time $2^{o(n)}$.*

Similar to NP-hardness, an ETH-lower bound is a way of connecting the hardness of a new and often poorly understood problem to problems we already have a good understanding of. The idea is to show that an improvement on the runtime of the currently best algorithm for a new problem would disprove the ETH. Although the ETH remains unproven, the continued absence of any algorithm for 3-SAT fast enough to disprove the ETH is itself strong empirical evidence in support of the hypothesis.

3 Overview of the algorithms

Our algorithms are all dynamic programs on a tree decomposition (T, X) of the Hasse diagram of a simplicial complex K . For each node $t \in T$, starting at the leaves of T and moving towards the root, we compute a set of candidate solutions to our problem, where a candidate solution is a subcomplex of K that might be a subcomplex of a solution to our problem. We recursively use candidate solutions at the children of t to build the candidate solutions at t . At the end of the algorithm, candidate solutions at the root of t will be solutions to our problem. In this section, we explore how a candidate solution to our problem is defined, and how we can effectively store representations of these candidate solutions so that our final algorithm is FPT.

Certain nice tree decompositions³ (T, X) of the Hasse diagram of a simplicial complex K define a recursively-nested set of subcomplexes of K . Recall that each bag of the tree decomposition is a set of simplices of K . For each node $t \in T$, the subcomplex $K_t \subset K$ is the union of the bags of each descendant of t minus the triangles in the bag of t . These subcomplexes have the property that if t' is a child of t , then $K_{t'} \subset K_t$.

We use this set of subcomplexes to recursively build solutions to our problems. Our algorithm computes a set of **candidate solutions** at each node t . The exact definition of candidate solution is given in Section 3.5 of the extended version of this paper [5], but

² One such example is sorting, which we know can at best be done in $\Omega(n \log n)$ time.

³ Certain here means “closed”, which is a type of tree decomposition of the Hasse diagram we define in Section 3.3 of the the extended version of this paper [5]. In particular, the set K_t as defined above is a simplicial complex in a closed tree decomposition, which is not true for general tree decompositions.

intuitively, a candidate solution at a node t is a subcomplex of K_t that could be a subcomplex of a combinatorial surface in K . In particular, the link of each vertex in a candidate solution must be a subset of a simple path or simple cycle. Our definition of candidate solution works recursively: if Σ is a candidate solution at t , then for each child t' of t , the complex $\Sigma \cap K_{t'}$ is a candidate solution at t' . Our algorithm uses this fact to find candidate solutions at t . Specifically, our algorithm attempts to build candidate solutions at t by growing candidate solutions at t' .

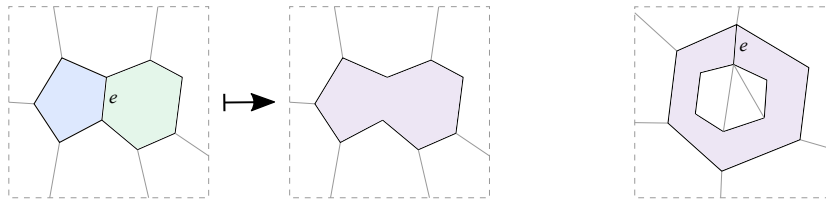
The main challenge with this approach is storing candidate solutions. There can be an exponential number of candidate solutions at a given node t , so we cannot simply store all candidate solutions. Generally, dynamic programs on tree decompositions work by storing some local representation of candidate solutions at t , where a local representation is a description of a candidate solution only in terms of vertices and edges in the bag X_t . Two candidate solutions with the same local representation are typically interchangeable in the sense that one candidate solution can be extended to a complete solution if and only if the other can too. The number of these local representations at t is typically a function of the size of X_t , which allows for FPT algorithms parameterized by the treewidth.

The local representation of candidate solutions for our problems should have several properties. First, they should represent a candidate solution using only simplices in X_t . Second, they should retain enough information that we can verify that a subcomplex is a candidate solution, i.e. it could be extended to a surface in K . In particular, we should be able to deduce information about the links of simplices in X_t from the local representation. The first and second properties are at odds, as even if a simplex σ is contained in X_t , the link of σ need not be contained in X_t . Finally, we should be able to deduce the homeomorphism class of a candidate solution from the local representation. Again, this property is at odds with the first property, as topological properties like the genus and orientability of a surface are global, not local, properties of a surface. One of our contributions is introducing a data structure to store local representations of candidate solution with each of these properties called the **annotated cell complex**.

A (non-annotated) **cell complex** is an algebraic representation of a surface that was originally introduced by Ahlfors and Sario [1] to prove the Classification Theorem of Compact Surfaces. Intuitively, a cell complex is a collection of disks, called **faces**, joined by shared edges in their boundaries. The faces in a cell complex differ from triangles in a simplicial complex as the faces in a cell complex can have more than three edges in their boundary. A definition of cell complex and a discussion of their properties can be found in Section 3.2 of the extended version of this paper [5].

The advantage of using cell complexes rather than simplicial complexes to store surfaces is that there is a simple equivalence relation that partitions cell complexes into homeomorphism classes. This is of obvious benefit as the surface S we are looking for may be specified by its homeomorphism class, but there is a secondary benefit. We define a set of **equivalence-preserving moves**, operations on cell complexes that preserve their homeomorphism class. We use these moves to compress the local representation of each candidate solution we keep during our algorithm. The most important benefit that these moves provide is the ability to merge two faces that share an edge.

To see why merging faces is helpful, suppose that we have a candidate solution Σ at a node t that is represented as a cell complex. We would like to store a local representation of Σ using only edges in X_t . There would then be a bounded number of local representations of candidate solutions at a node t , as there are a bounded number of edges in X_t . To this end, each time we forget an edge e , we would like to merge the two faces incident to e into a single face. See Figure 4, left panel.



■ **Figure 4** Left: The edge e is removed by merging the two incident faces. Right: The edge e appears twice on the boundary of the same face, so e cannot be removed by merging incident faces as this would make the interior of the face an annulus. We use annotated cell complexes to remove e .

The idea of merging faces when we forget e works unless e is incident to the same face twice; the right panel of Figure 4 gives an example. After merging some faces, it is possible that a face may have two edges on its boundary identified. If two edges on the boundary of the same face are identified, then we can no longer remove these edges by merging their incident faces, as then the interior of this face would no longer be a disk.

We therefore modify the definition of cell complex to allow for a more general type of face. Our first change is to allow a face to be a disk with multiple boundary components like in Figure 4, but we need to go a step further. Topological features like handles, crosscaps, and boundaries in cell complexes are the result of a single face having edges on its boundary identified in certain ways; thus, we need a way of removing the edges that constitute these topological features. An **annotated cell complex** annotates each face with the number of topological features like handles, crosscaps, and boundaries on this face, rather than storing these features explicitly with edges. In effect, an annotated cell complex is a representation of a surface where the interior of a face is allowed to be any compact connected surface.

4 Overview of the lower bounds

Here we present the main ideas that go into the proof of our lower bounds. The omitted details can be found in Section 4 of the extended version of this paper [5].

► **Theorem 2.** *Assuming the ETH, no algorithm can solve SUBSURFACE RECOGNITION, SUM-OF-GENUS SUBSURFACE RECOGNITION or SUBSURFACE PACKING in $2^{o(k \log k)} n^{O(1)}$ time. The parameter k denotes the width of a given (nice) path decomposition of the Hasse diagram of the input simplicial complex.*

Since every path decomposition is also a tree decomposition, the treewidth of a graph is never higher than its pathwidth. Theorem 2 therefore implies that none of our problems can be solved in $2^{o(k \log k)} n^{O(1)}$ time, where k is now the treewidth of the Hasse diagram.

We focus on proving the result for SUBSURFACE RECOGNITION. After this, it will be easy to modify our arguments to prove similar results for the two other problems. At a conceptual level there are two parts to the proof.

1. Define a reduction from DIRECTED CYCLE PACKING to SUBSURFACE RECOGNITION.
2. Show that the reduction can always be chosen so that the pathwidth of the output space is bounded by some linear function of the pathwidth of the input graph.

4.1 The reduction

DIRECTED CYCLE PACKING asks us to find as many vertex disjoint cycles in a graph as possible (see Figure 5). This problem is essentially a directed, 1-dimensional version of the SP problem, since we know that the only compact 1-manifolds are circles (cycles) and closed intervals (paths).

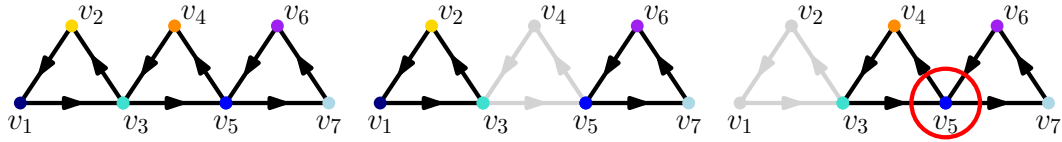
17:10 ETH-Tight Algorithms for Finding Surfaces

► **Problem 5.** The *DIRECTED CYCLE PACKING (DCP)* problem

INPUT: A directed graph D on n vertices and an integer ℓ .

PARAMETER: The pathwidth k of D .

QUESTION: Does D contain ℓ vertex disjoint cycles?



■ **Figure 5** A directed graph D (left), two vertex disjoint cycles contained in D (middle) and two cycles in D intersecting at a common vertex (right). This will be a guiding example for this section.

The DCP problem is a good starting point for our reduction not only because of its similarity to the SP problem but also because of the following theorem.

► **Theorem 3** ([17]). *Assuming the ETH, the DCP problem cannot be solved in $2^{o(k \log k)} n^{O(1)}$ time, where the parameter k denotes the width of a given (nice) path decomposition of the input graph.*

Given a digraph D , the reduction will construct a 2-dimensional simplicial complex Y that contains ℓ disjoint tori if and only if D contains ℓ vertex disjoint cycles. In fact, we show that the only connected subsurfaces without boundary in Y are tori and that these are in a bijection with the directed cycles in D . Furthermore, any pair of these tori are *disjoint* if and only if the corresponding directed cycles are vertex disjoint.

In Figure 6 we introduce some shorthand notation that will help make the reduction clearer. Each column of the figure shows a different component that we will use when constructing the space Y . The first row shows the shorthand notation. The second row shows the “topological space” that the notation represents. The third and fourth row indicate which triangulation we use to represent this space.

The first column shows a cylinder, S_1 . The second column shows a space S_2 consisting of two cylinders, X'_1 and X'_2 . These cylinders are glued together at a single interior point, called a (0-dimensional) **singularity**. The third column shows a space S_3 consisting of three cylinders X''_1 , X''_2 and X''_3 , each with a single boundary component attached to the same circle. The fourth and final column shows the space S_4 , obtained by gluing S_2 and S_3 together. More precisely, S_4 also consists of three cylinders, $X_1 = X'_1 \cup X''_1$, $X_2 = X'_2 \cup X''_2$ and $X_3 = X''_3$, each having a single boundary component attached to the same circle. Additionally, $X_1 \cup X_2$ contains a 0-dimensional singularity.

We establish some important properties of the spaces S_1, S_2, S_3 and S_4 from Figure 6. In order to describe these properties we temporarily extend the notion of a “boundary”, a term usually reserved for manifolds, to the world of simplicial complexes. In the remainder of this section, the word boundary will refer to the closure of the set of 1-simplices in X that only have a single coface. We denote this subcomplex by $\partial(X)$.

► **Remark 4.** Let S_1, S_2, S_3 and S_4 be the spaces introduced in Figure 6.

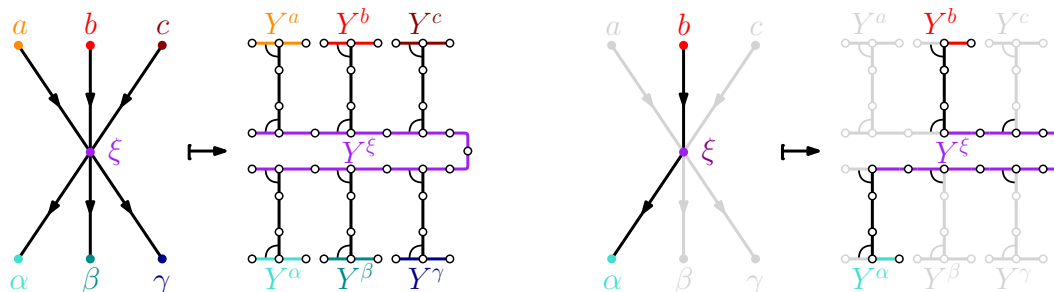
1. The only (non-empty) 2-manifold $X \subseteq S_1$ where $\partial(X) \subseteq \partial(S_1)$ is S_1 itself.
2. The only 2-manifolds $X \subseteq S_2$ where $\partial(X) \subseteq \partial(S_2)$ are X'_1 and X'_2 .
3. The only 2-manifolds $X \subseteq S_3$ where $\partial(X) \subseteq \partial(S_3)$ are $X''_1 \cup X''_2$, $X''_1 \cup X''_3$ and $X''_2 \cup X''_3$.
4. The only 2-manifolds $X \subseteq S_4$ where $\partial(X) \subseteq \partial(S_4)$ are $X_1 \cup X_3$ and $X_2 \cup X_3$.

Name	S_1	S_2	S_3	S_4
Notation				
Space		X'_1 X'_2 	X''_1 X''_2 X''_3 	X_1 X_2 X_3
Simplicial Complex				
Detailed & Unfolded Simplicial Complex				

■ **Figure 6** Shorthand notation for specific triangulations of S_1, \dots, S_4 that we will use frequently throughout the section.

That these properties holds is intuitively obvious. Formally, this can be proved easily by brute force: Simply go through all the 2-simplices in S_i and assume that it is contained in a submanifold X . It is then easy to see which adjacent 2-simplices must necessarily also be contained in the same submanifold. Whenever a choice has to be made, simply branch and try all possibilities.

The reduction is perhaps best understood in terms of vertex gadgets and edge gadgets. In particular, Figure 7 shows how a vertex ξ is mapped to the vertex gadget Y^ξ , using the notation from Figure 6. The figure also shows six edge gadgets (in black), three corresponding to the edges entering ξ and three corresponding to the edges leaving ξ . The edge gadgets are unlabeled in the figure but can be identified by the vertex gadgets they are attached to. We think of each vertex gadget as composed of two sub-cylinders, one half for the incoming edge gadgets and the other half for outgoing edge gadgets. To better see this separation we draw the vertex gadget with a U-turn at the location of this divide in our figures.

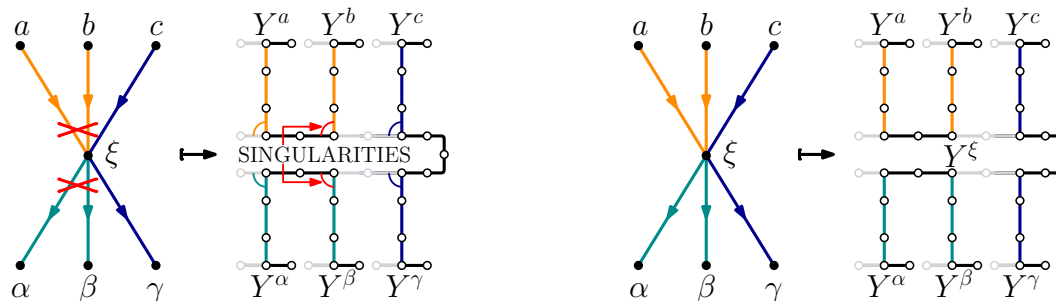


■ **Figure 7** A local view of how a vertex ξ is mapped to its vertex gadget Y^ξ (left) and an illustration of how a directed cycle passing through the vertex ξ is mapped to a submanifold in the space (right).

17:12 ETH-Tight Algorithms for Finding Surfaces

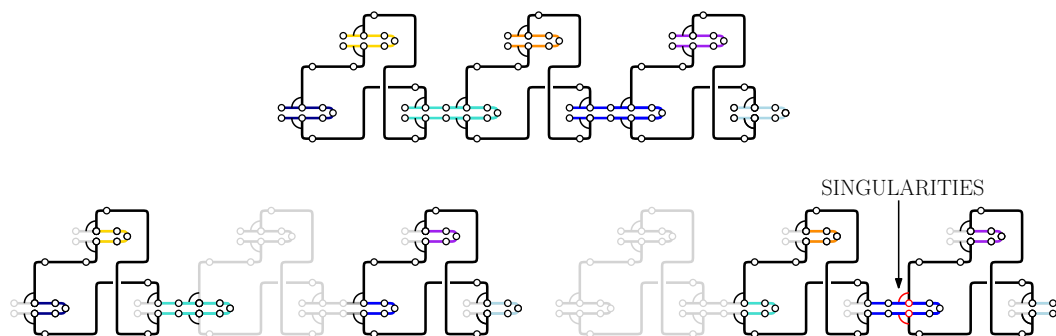
Each edge gadget is connected to the vertex gadgets corresponding to each of its two ends through a copy of S_4 . The edge gadget contains the cylinder X_1 while the vertex gadget contains the other cylinders X_2 and X_3 . Both the incoming and outgoing part of the vertex gadget consists primarily of a sequence of smaller cylinders, $X_2 \cup X_3$, one for each incoming/outgoing edge. The boundary of the X_3 corresponding to one edge is attached to the boundary of the copy of X_2 corresponding to the next edge. The boundary of the “last” X_3 of the incoming edges is attached to one boundary component of a single additional cylinder, while the “last” X_3 of the outgoing edges is attached to the other boundary component.

By repeated use of property 4, any potential manifold contained in this space must contain precisely one incoming and one outgoing edge gadget per vertex, assuming the manifold is not allowed to have a boundary. This is illustrated in Figure 8. This figure also shows the importance of the 0 dimensional singularities in the reduction. The resulting space could otherwise contain tori that do not correspond to any directed cycle. An example of the correspondence between disjoint tori and vertex disjoint directed cycles is shown in Figure 9.



■ **Figure 8** The leftmost figure shows how the singularities keeps “badly behaved” subcomplexes from becoming manifolds. The rightmost figure shows how the reduction would fail without the use of singularities between the vertex gadgets and edge gadgets.

We see in Figure 9 that we can associate any pair of vertex disjoint cycles in the input graph to a pair of non-intersecting tori in the output space in an obvious way. Concretely, a cycle is mapped to a torus by sending the edges to edge gadgets and by then connecting these through the vertex gadgets. This association turns out to be a bijection with an inverse that maps a submanifold to the set of edges whose edge gadgets intersects the submanifold. That this inverse is well-defined is proved for the pathwidth-preserving reduction in Section 4.5 of the extended version of the paper; see [5].

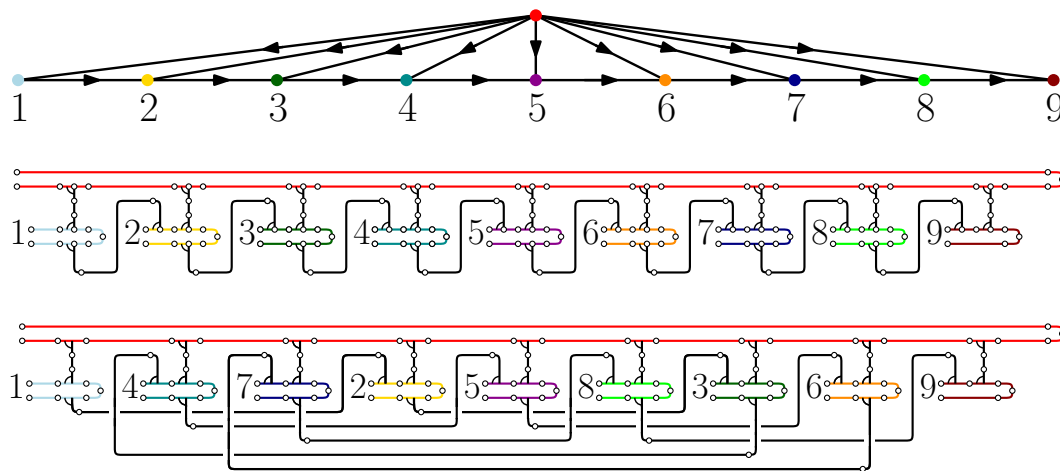


■ **Figure 9** An illustration of how the graph from Figure 5 is mapped to spaces and how valid/invalid subsets of edges are mapped to manifolds/non-manifolds respectively.

4.2 Pathwidth preservation

The main idea of this section can be summarized in a single sentence: By carefully choosing the order in which we attach edge/vertex gadgets to each other, we can make a space that has a similar structure to a nice path decomposition of the input graph. This is an absolutely necessary “fine tuning” of the reduction we saw in the previous subsection. Without it, we have no guarantee that the Hasse diagram of the space we construct will have low pathwidth. In fact, if the ordering is chosen in an adversarial way, we may end up mapping a graph of bounded pathwidth to spaces whose Hasse diagram has arbitrarily large pathwidth.

We discuss this in detail in Section 4.4.1 of the extended version of the paper [5], but the rough idea is captured in Figure 10. Here we see a graph of pathwidth 2 being mapped to two very different spaces. While both are constructed in a way that is compatible with the reduction described in Section 4.1, intuitively it is the topmost space that has retained most of the “pathlikeness” of the input graph. This intuition is reflected in the fact that the Hasse diagram of the lower figure really is higher than that of the one above it.

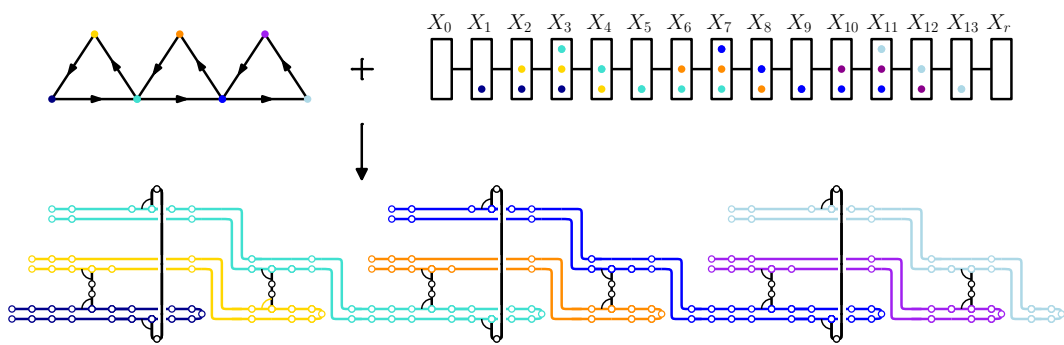


■ **Figure 10** A directed graph of pathwidth 2 (top) together with a “sensible” version of the reduction explained in Section 4.1 (middle) and an “adversarial” version of the reduction (bottom).

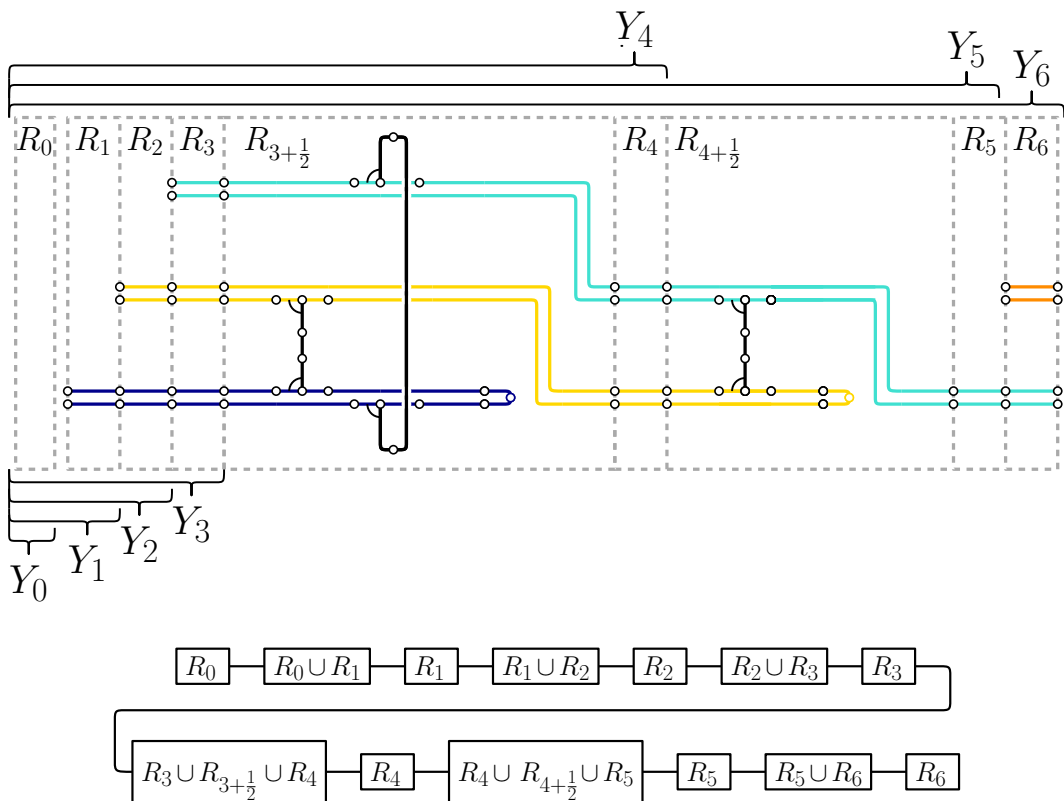
It can be quite hard to prove lower bounds on the path-/treewidth of graphs/spaces, but for this particular family it is reasonably straightforward. Once you generalize the above “adversarial” layout for any input graph on $n^2 + 1$ vertices there is a nice geometric argument that the Hasse diagram of the outputted space will always contain an $n \times n$ -grid as a graph minor. It is well known that such graphs have treewidth at least n which gives us our desired lower bound.

If we are given a less structured graph than the one we saw in Figure 10, it might be hard to see how we can best glue the gadgets together. Our way around this is to construct a space where the order in which vertex gadgets are attached to each other is determined by the order in which the nodes are forgotten in the nice path decomposition of the input graph. The idea is that a vertex gadget is attached to a neighbouring vertex gadget in the current bag when it (or its neighbour) is forgotten, see Figure 11.

The way we make the above idea precise is rather technical. It is in essence all about structural induction over the given nice path decomposition, which we use to construct a nested sequence of spaces $Y_0 \subset \dots \subset Y_r$, one for each bag. We also compute an accompanying path decomposition for the Hasse diagram of each of the nested spaces. These path decompositions



■ **Figure 11** An illustration of how the graph from Figure 5 (top left) is mapped to a space (bottom) having the same “structure”/“order” as the given nice path decomposition (top right) of the graph.



■ **Figure 12** The space Y_6 (top) associated to bag X_6 in the nice path decomposition of the graph in Figure 11. The location of the sub complexes $Y_0 \subset \dots \subset Y_5$ are indicated. Below is the path decomposition of Y_6 . Path decomposition of the other spaces $Y_i, 0 \leq i \leq 5$ are all present as the path decomposition induced by “sub-paths” starting at the bag containing R_0 and ending at the bag containing R_i .

are not optimal, but their width is bounded above by the width of the inputted nice path decomposition times a constant, which is sufficient for our purposes. The induction involves going through a lot of elementary claims about the space we have constructed at each step. For details on this, see Section 4.4.2 of the extended version of this paper [5]. The space Y_6 and its path decomposition are shown in Figure 12.

5 Conclusion

In this paper, we consider the parameterized complexity of several variants of the problem of finding surfaces in 2-dimensional simplicial complexes with respect to the treewidth of the Hasse diagram. We give ETH-optimal algorithms for the SUM-OF-GENUS SUBSURFACE RECOGNITION and SUBSURFACE PACKING problems. We also give an ETH-based lower bound for Subsurface Recognition and an FPT algorithm for CONNECTED SUBSURFACE RECOGNITION. Several questions surrounding subsurface recognition remain open, such as

- whether the algorithm presented in this paper for CONNECTED SUBSURFACE RECOGNITION is ETH-optimal;
- whether or not the Subsurface Recognition Problem is $W[1]$ -hard when parameterized by the treewidth of the Hasse diagram.

Future work could either attempt to find better parameterized algorithms or prove stronger lower bounds for these problems.

References

- 1 L.V. Ahlfors and L. Sario. *Riemann Surfaces*. Princeton mathematical series. Princeton University Press, 2015. URL: <https://books.google.com/books?id=4C4PAAAAIAAJ>.
- 2 Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k -tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987. doi:10.1137/0608024.
- 3 László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 684–697, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897542.
- 4 Bhaskar Bagchi, Basudeb Datta, Benjamin A. Burton, Nitin Singh, and Jonathan Spreer. Efficient Algorithms to Decide Tightness. In Sándor Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry (SoCG 2016)*, volume 51 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:15, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.SoCG.2016.12.
- 5 Mitchell Black, Nello Blaser, Amir Nayyeri, and Erlend Raa Vågset. ETH-tight algorithms for finding surfaces in simplicial complexes of bounded treewidth. *CoRR*, abs/2203.07566, 2022. arXiv:2203.07566.
- 6 Nello Blaser, Morten Brun, Lars M. Salbu, and Erlend Raa Vågset. The parameterized complexity of finding minimum bounded chains. *CoRR*, 2021. arXiv:2108.04563.
- 7 Nello Blaser and Erlend Raa Vågset. Homology localization through the looking-glass of parameterized complexity theory, 2020. arXiv:2011.14490.
- 8 Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michał Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016. doi:10.1137/130947374.
- 9 Benjamin Burton, Sergio Cabello, Stefan Kratsch, and William Pettersson. The Parameterized Complexity of Finding a 2-Sphere in a Simplicial Complex. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science (STACS 2017)*, volume 66 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.STACS.2017.18.
- 10 Benjamin Burton and Rodney Downey. Courcelle’s theorem for triangulations. *Journal of Combinatorial Theory, Series A*, 146, March 2014. doi:10.1016/j.jcta.2016.10.001.
- 11 Benjamin A. Burton, Thomas Lewiner, João Paixão, and Jonathan Spreer. Parameterized complexity of discrete Morse theory. *ACM Transactions on Mathematical Software*, 42(1):1–24, March 2016. doi:10.1145/2738034.

- 12 Benjamin A. Burton and Jonathan Spreer. The complexity of detecting taut angle structures on triangulations. *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2013. doi:10.1137/1.9781611973105.13.
- 13 A.V. Chernavsky and V.P. Leksine. Unrecognizability of manifolds. *Annals of Pure and Applied Logic*, 141(3):325–335, 2006. Papers presented at the Second St. Petersburg Days of Logic and Computability Conference on the occasion of the centennial of Andrey Andreevich Markov, Jr. doi:10.1016/j.apal.2005.12.011.
- 14 Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71*, pages 151–158, New York, NY, USA, 1971. Association for Computing Machinery. doi:10.1145/800157.805047.
- 15 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.
- 16 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer International Publishing, 2015. doi:10.1007/978-3-319-21275-3.
- 17 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 150–159. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.23.
- 18 Wolfgang Haken. Theorie der normalflächen. *Acta Mathematica*, 105(3):245–375, September 1961. doi:10.1007/BF02559591.
- 19 Sergei Ivanov. Computational complexity. MathOverflow. URL: <https://mathoverflow.net/q/118428>.
- 20 A. Markov. The insolubility of the problem of homeomorphy. *Dokl. Akad. Nauk USSR*, 12(2):218–220, 1958.
- 21 Hyam Rubinstein. The solution to the recognition problem for \mathbb{S}^3 . Lecture, 1992.
- 22 Abigail Thompson. Thin position and the recognition problem for \mathbb{S}^3 . *Mathematical Research Letters*, 1(5):613–630, 1994.

Asymptotic Bounds on the Combinatorial Diameter of Random Polytopes

Gilles Bonnet ✉

University of Groningen, The Netherlands

Daniel Dadush ✉

Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Uri Grupel ✉

Universität Innsbruck, Austria

Sophie Huiberts ✉

Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Galyna Livshyts ✉

Georgia Institute of Technology, Atlanta, GA, USA

Abstract

The combinatorial diameter $\text{diam}(P)$ of a polytope P is the maximum shortest path distance between any pair of vertices. In this paper, we provide upper and lower bounds on the combinatorial diameter of a random “spherical” polytope, which is tight to within one factor of dimension when the number of inequalities is large compared to the dimension. More precisely, for an n -dimensional polytope P defined by the intersection of m i.i.d. half-spaces whose normals are chosen uniformly from the sphere, we show that $\text{diam}(P)$ is $\Omega(nm^{\frac{1}{n-1}})$ and $O(n^2m^{\frac{1}{n-1}} + n^54^n)$ with high probability when $m \geq 2^{\Omega(n)}$.

For the upper bound, we first prove that the number of vertices in any fixed two dimensional projection sharply concentrates around its expectation when m is large, where we rely on the $\Theta(n^2m^{\frac{1}{n-1}})$ bound on the expectation due to Borgwardt [Math. Oper. Res., 1999]. To obtain the diameter upper bound, we stitch these “shadows paths” together over a suitable net using worst-case diameter bounds to connect vertices to the nearest shadow. For the lower bound, we first reduce to lower bounding the diameter of the dual polytope P° , corresponding to a random convex hull, by showing the relation $\text{diam}(P) \geq (n-1)(\text{diam}(P^\circ) - 2)$. We then prove that the shortest path between any “nearly” antipodal pair vertices of P° has length $\Omega(m^{\frac{1}{n-1}})$.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Random Polytopes, Combinatorial Diameter, Hirsch Conjecture

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.18

Related Version *Full Version:* <https://arxiv.org/abs/2112.13027>

Funding *Gilles Bonnet:* Funded by the DFG Priority Program (SPP) 2265 Random Geometric Systems, project P23.

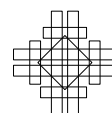
Daniel Dadush: Supported by the ERC Starting grant QIP-805241.

Acknowledgements This work was done in part while the authors were participating in the Probability, Geometry and Computation in High Dimensions semester at the Simons Institute for the Theory of Computing and in the Interplay between High-Dimensional Geometry and Probability trimester at the Hausdorff Institute for Mathematics.



© Gilles Bonnet, Daniel Dadush, Uri Grupel, Sophie Huiberts, and Galyna Livshyts; licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 18; pp. 18:1–18:15
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

When does a polyhedron have small (combinatorial) diameter? This question has fascinated mathematicians, operation researchers and computer scientists for more than half a century. In a letter to Dantzig in 1957, motivated by the study of the simplex method for linear programming, Hirsch conjectured that any n -dimensional polytope with m facets has diameter at most $m - n$. While recently disproved by Santos [29] (for unbounded polyhedra, counter-examples were already given by Klee and Walkup [20]), the question of whether the diameter is bounded from above by a polynomial in n and m , known as the *polynomial Hirsch conjecture*, remains wide open. In fact, the current counter-examples violate the conjectured $m - n$ bound by at most 25 percent.

The best known general upper bounds on the combinatorial diameter of polyhedra are the $2^{n-3}m$ bound by Barnette and Larman [3, 23, 4], which is exponential in n and linear in m , and the *quasi-polynomial* $m^{\log_2 n+1}$ bound by Kalai and Kleitman [19]. The Kalai-Kleitman bound was recently improved to $(m-n)^{\log_2 n}$ by Todd [33] and $(m-n)^{\log_2 O(n/\log n)}$ by Sukegawa [32]. Similar diameter bounds have been established for graphs induced by certain classes of simplicial complexes, which vastly generalize 1-skeleta of polyhedra. In particular, Eisenbrand et al [16] proved both Barnette-Larman and Kalai-Kleitman bounds for so-called connected-layer families (see Theorem 20), and Labbé et al [22] extended the Barnette-Larman bound to pure, normal, pseudo-manifolds without boundary.

Moving beyond the worst-case bounds, one may ask for which families of polyhedra does the Hirsch conjecture hold, or more optimistically, are there families for which we can significantly beat the Hirsch conjecture? Many interesting classes induced by combinatorial optimization problems are known, including the class of polytopes with vertices in $\{0, 1\}^n$ [25], Leontief substitution systems [18], transportation polyhedra and their duals [1, 10, 9], as well as the fractional stable-set and perfect matching polytopes [24, 28].

Relatedly, there has been progress on obtaining diameter bounds for classes of “well-conditioned” polyhedra. If P is a polytope defined by an integral constraint matrix $A \in \mathbb{Z}^{m \times n}$ with all square submatrices having determinant of absolute value at most Δ , then diameter bounds polynomial in m, n and Δ have been obtained [15, 5, 11, 26]. The best current bound is $O(n^3 \Delta^2 \log(\Delta))$, due to [11]. Extending on the result of Naddef [25], strong diameter bounds have been proved for polytopes with vertices in $\{0, 1, \dots, k\}^n$ [21, 13, 14]. In particular, [21] proved that the diameter is at most nk , which was improved to $nk - \lceil n/2 \rceil$ for $k \geq 2$ [13] and to $nk - \lceil 2n/3 \rceil - (k-2)$ for $k \geq 4$ [14].

1.1 Diameter of Random Polytopes

With a view of beating the Hirsch bound, the main focus of this paper will be to analyze the diameter of random polytopes, which one may think of as well-conditioned on “average”. Coming both from the average case and smoothed analysis literature [6, 7, 31, 34, 12], there is tantalizing evidence that important classes of random polytopes may have very small diameters.

In the average-case context, Borgwardt [6, 7] proved that for $P(A) := \{x \in \mathbb{R}^n : Ax \leq 1\}$, $A \in \mathbb{R}^{m \times n}$, where the rows of A are drawn from any rotational symmetric distribution (RSD), that the expected number of edges in any fixed 2 dimensional projection of P – the so-called *shadow bound* – is $O(n^2 m^{\frac{1}{n-1}})$. Borgwardt also showed that this bound is tight up to constant factors when the rows of A are drawn uniformly from the sphere, that is, the expected shadow size is $\Theta(n^2 m^{\frac{1}{n-1}})$. In the smoothed analysis context, A has the form $\bar{A} + \sigma G$, where \bar{A} is a fixed matrix with rows of ℓ_2 norm at most 1 and G has i.i.d. standard

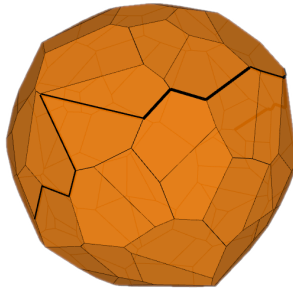
normally distributed entries and $\sigma > 0$. Bounds on the expected size of the shadow in this context were first studied by Spielman and Teng [31], later improved by [34, 12], where the best current bound is $O(n^2\sqrt{\log m}/\sigma^2)$ due to [12] when $\sigma \leq \frac{1}{\sqrt{n \log m}}$.

From the perspective of short paths, these results imply that if one samples objectives v, w uniformly from the sphere, then there is a path between the vertices maximizing v and w in P of expected length $O(n^2 m^{\frac{1}{n-1}})$ in the RSD model, and expected length $O(n^2\sqrt{\log m}/\sigma^2)$ in the smoothed model. That is, “most pairs” of vertices (with respect to the distribution in the last sentence), are linked by short expected length paths. Note that both of these bounds scale either sublinearly or logarithmically in m , which is far better than $m - n$. While these bounds provide evidence, they do not directly upper bound the diameter, since this would need to work for all pairs of vertices rather than most pairs.

A natural question is thus whether the shadow bound is close to the true diameter. In this paper, we show that this is indeed the case, in the setting where the rows of A are drawn uniformly from the sphere and when m is (exponentially) large compared to n . More formally, our main result is as follows:

► **Theorem 1.** *Suppose $n, m \in \mathbb{N}$ satisfy $n \geq 2$ and $m \geq 2^{\Omega(n)}$. Let $A^\top := (a_1, \dots, a_M) \in \mathbb{R}^{n \times M}$, where M is Poisson distributed with $\mathbb{E}[M] = m$, and a_1, \dots, a_M are sampled independently and uniformly from \mathbb{S}^{n-1} . Then, letting $P(A) := \{x \in \mathbb{R}^n : Ax \leq 1\}$, with probability at least $1 - m^{-n}$, we have that*

$$\Omega(nm^{\frac{1}{n-1}}) \leq \text{diam}(P(A)) \leq O(n^2 m^{\frac{1}{n-1}} + n^5 4^n).$$



■ **Figure 1** A diameter achieving path for a random spherical polytope with 100 constraints.

In the above, we note that the number of constraints M is chosen according to a Poisson distribution with expectation m . This is only for technical convenience (it ensures useful independence properties) and with small modifications, our arguments also work in the case where $M := m$ deterministically. Also, since the constraints are chosen from the sphere, M is almost surely equal to the number of facets of $P(A)$ above (i.e., there are no redundant inequalities).

From the bounds, we see that $\text{diam}(P(A)) \leq O(n^2 m^{\frac{1}{n-1}})$ with high probability as long as $m \geq 2^{\Omega(n^2)}$. This shows that the shadow bound is indeed close to an upper bound for the expected diameter when m is sufficiently large. Furthermore, the shadow bound is tight to within one factor of dimension in this regime. We note that the upper bound is already non-trivial when $m \geq \Omega(n^5 4^n)$, since then $O(n^2 m^{\frac{1}{n-1}} + n^5 4^n) \leq m - n$.

While our bounds are only interesting when m is exponential, the bounds are nearly tight asymptotically, and as far as we are aware, they represent the first non-trivial improvements over worst-case upper bounds for a natural class of polytopes defined by random halfspaces.

Our work naturally leaves two interesting open problems. The first is whether the shadow bound upper bounds the diameter when m is polynomial in n . The second is to close the factor n gap between upper and lower bound in the large m regime.

1.2 Prior work

Lower bounds on the diameter of $P(A)$, $A^\top = (a_1, \dots, a_m) \in \mathbb{R}^{n \times m}$, were studied by Borgwardt and Huhn [8]. They examined the case where each row of A is sampled from a RSD with radial distribution $\Pr_a[\|a\|_2 \leq r] = \frac{\int_0^r (1-t^2)^\beta t^{n-1} dt}{\int_0^1 (1-t^2)^\beta t^{n-1} dt}$, for $r \in [0, 1]$, $\beta \in (-1, \infty)$. Restricting their results to the case $\beta \rightarrow -1$, corresponding to the uniform distribution on the sphere (where the bound is easier to state), they show that $\mathbb{E}[\text{diam}(P(A))] \geq \Omega(m^{\frac{1}{n} + \frac{1}{n(n-1)^2}})$. We improve their lower bound to $\Omega(nm^{1/(n-1)})$ when $m \geq 2^{\Omega(n)}$, noting that $m^{1/(n-1)} = O(1)$ for $m = 2^{O(n)}$.

In terms of upper bounds, the diameter of a *random convex hull of points*, instead of a random intersection of halfspaces, has been implicitly studied. Given $A^\top = (a_1, \dots, a_m) \in \mathbb{R}^{n \times m}$, let us define

$$Q(A) := \text{conv}(\{a_1, \dots, a_m\}) \tag{1}$$

to be the convex hull of the rows of A . When the rows of A are sampled uniformly from \mathbb{B}_2^n , the question of when the diameter of $Q(A)$ is exactly 1 (i.e., every pair of distinct vertices is connected by an edge) was studied by Bárány and Füredi[2]. They proved that with probability $1 - o(1)$, $\text{diam}(Q(A)) = 1$ if $m \leq 1.125^n$ and $\text{diam}(Q(A)) > 1$ if $m \geq 1.4^n$.

In dimension 3, letting $a_1, \dots, a_M \in \mathbb{S}^2$ be chosen independently and uniformly from the 2-sphere, where M is Poisson distributed with $\mathbb{E}[M] = m$, Glisse, Lazard, Michel and Pouget [17] proved that with high probability the maximum number of edges in any 2-dimensional projection of $Q(A)$ is $\Theta(\sqrt{m})$. This in particular proves that the combinatorial diameter is at most $O(\sqrt{m})$ with high probability.

It is important to note that the geometry of $P(A)$ and $Q(A)$ are strongly related. Indeed, as long as $m = \Omega(n)$ and the rows of A are drawn from a symmetric distribution, $P(A)$ and $Q(A)$ are polars of each other. That is, $Q(A)^\circ = P(A)$ and $P(A)^\circ := \{x \in \mathbb{R}^n : \langle x, y \rangle \leq 1, \forall y \in P(A)\} = Q(A)^1$.

As we will see, our proof of Theorem 1 will in fact imply similarly tight diameter bounds for $\text{diam}(Q(A))$ as for $\text{diam}(P(A))$, yielding analogues and generalizations of the above results, when $A^\top = (a_1, \dots, a_M) \in \mathbb{R}^{n \times M}$ and M is Poisson with $\mathbb{E}[M] = m$. More precisely, we will show that for $m \geq 2^{\Omega(n)}$, with high probability

$$\Omega(m^{\frac{1}{n-1}}) \leq \text{diam}(Q(A)) \leq O(nm^{\frac{1}{n-1}} + n^5 4^n).$$

In essence, for m large enough, our bounds for $\text{diam}(Q(A))$ are a factor $\Theta(n)$ smaller than our bounds for $\text{diam}(P(A))$. This relation will be explained in Section 4.

1.3 Proof Overview

In this section, we give the high level overview of our approach.

¹ Precision: $P(A) = Q(A)^\circ$ always holds and $P(A)^\circ = Q(A)$ requires that $0 \in Q(A)$ which, as a direct consequence of Wendel's theorem [30, Theorem 8.2.1], happens with probability $1 - o(1)$ when $m \geq cn$ for any fixed $c > 2$. In general $P(A)^\circ = \text{conv}(A \cup \{0\})$ holds.

1.3.1 The Upper Bound

In this overview, we will say that an event holds with high probability if it holds with probability $1 - m^{-\Omega(n)}$. To prove the upper bound on the diameter of $P(A)$, we proceed as follows. For simplicity, we will only describe the high level strategy for achieving a $O(n^2 m^{\frac{1}{n-1}} + 2^{O(n)})$ bound. To begin, we first show that the vertices of $P(A)$ maximizing objectives in a suitable net N of the sphere \mathbb{S}^{n-1} , are all connected to the vertex maximizing e_1 , with a path of length $O(n^2 m^{\frac{1}{n-1}} + 2^{O(n)})$ with high probability. Second, we will show that with high probability, for all $v \in \mathbb{S}^{n-1}$, there is a path between the vertex of $P(A)$ maximizing v and the corresponding maximizer of closest objective $v' \in N$ of length at most $2^{O(n)} \log m$. Since every vertex of $P(A)$ maximizes some objective in \mathbb{S}^{n-1} , by stitching at most 4 paths together, we get that the diameter of $P(A)$ is at most $O(n^2 m^{\frac{1}{n-1}} + 2^{O(n)} \log m) = O(n^2 m^{\frac{1}{n-1}} + 2^{O(n)})$ with high probability.

We only explain the strategy for the first part, as the second part follows easily from the same techniques. The key estimate here is the sharp $\Theta(n^2 m^{\frac{1}{n-1}})$ bound on the expected number of vertices in a fixed two dimensional projection due to Borgwardt [6, 7], the so-called *shadow bound*, which allows one to bound the expected length of paths between vertices maximizing any two fixed objectives (see Section 3 for a more detailed discussion). We first strengthen this result by proving that the size of the shadow sharply concentrates around its expectation when m is large (Theorem 9), allowing us to apply a union bound on a suitable net of shadows, each corresponding to a two dimensional plane spanned by e_1 and some element of N above. To obtain such concentration, we show that the shadow decomposes into a sum of nearly independent “local shadows”, corresponding to the vertices maximizing a small slice of the objectives in the plane, allowing us to apply concentration results on sums of nearly independent random variables.

Independence via Density

We now explain the local independence structure in more detail. For this purpose, we examine the smallest $\epsilon > 0$ such that rows of A are ϵ -dense on \mathbb{S}^{n-1} , that is, such that every point in \mathbb{S}^{n-1} is at distance at most ϵ from some row of A . Using standard estimates on the measure of spherical caps and the union bound, one can show with high probability that $\epsilon := \Theta((\log m/m)^{1/m})$ and that any spherical cap of radius $t\epsilon$ contains at most $O(t^{n-1} \log m)$ rows of A for any fixed $t \geq 1$.

We derive local independence from the fact that the vertex v of $P(A)$ maximizing a unit norm objective w is defined by constraints $a \in A$ which are distance at most 2ϵ from w (see Lemma 15 for a more general statement). This locality implies that the number of vertices in a projection of $P(A)$ onto a two dimensional subspace $W \ni w$ maximizing objectives at distance ϵ from w (i.e., the slice of objectives) depends only on the constraints in A at distance at most $O(\epsilon)$ from w . In particular, the number of relevant constraints for all objectives at distance ϵ from w is at most $2^{O(n)} \log m$ by the estimate in the last paragraph. By the independence properties of Poisson processes, one can in fact conclude that this local part of the shadow on W is independent of the constraints in A at distance more than $O(\epsilon)$ from w .

Given the above, we decompose the shadow onto W into $k = O(1/\epsilon)$ pieces, by placing k equally spaced objectives $w_0, \dots, w_{k-1}, w_k = w_0$ on $\mathbb{S}^{n-1} \cap W$, so that $\|w_i - w_{i+1}\|_2 \leq \epsilon$, $0 \leq i \leq k-1$, and defining $K_i \geq 0$, $0 \leq i \leq k-1$, to be the number of vertices maximizing objectives in $[w_i, w_{i+1}]$. This subdivision partitions the set of shadow vertices, so Borgwardt’s bound applies to the expected sum: $\mathbb{E}[\sum_{i=0}^{k-1} K_i] = O(n^2 m^{1/(n-1)})$. Furthermore, as argued

above, each K_i is (essentially) independent of all K_j 's with $|i - j \bmod k| = \Omega(1)$. This allows us to apply a Bernstein-type concentration bound for sums of nearly-independent bounded random variables to $\sum_{i=0}^{k-1} K_i$.

Unfortunately, the worst-case upper bounds we have for each K_i , $0 \leq i \leq k-1$, are rather weak. Namely, we only know that in the worst-case, K_i is bounded by the total number of vertices induced by constraints relevant to the interval $[w_i, w_{i+1}]$, where $\|w_i - w_{i+1}\| \leq \epsilon$. As mentioned above, the number of relevant constraints is $2^{O(n)} \log m$ and hence the number of vertices is at most $(2^{O(n)} \log m)^n$. With these estimates, we can show high probability concentration of the shadow size around its mean when $m \geq 2^{\Omega(n^3)}$. One important technical aspect ignored above is that both the independence properties and the worst-case upper bounds on each K_i crucially relies only on conditioning A to be “locally” ϵ -dense around $[w_i, w_{i+1}]$ (see Definition 16 and Lemma 19 for more details).

Abstract Diameter Bounds to the Rescue

To allow tight concentration to occur for $m = 2^{\Omega(n^2)}$, we adapt the above strategy by successively following shortest paths instead of the shadow path on W . More precisely, between the maximizer v_i of w_i and v_{i+1} of w_{i+1} , $0 \leq i \leq k-1$, we follow the shortest path from v_i to v_{i+1} in the subgraph induced by the vertices v of $P(A)$ satisfying $\langle v, w_{i+1} \rangle \geq \langle v_i, w_{i+1} \rangle$. We now let K_i , $0 \leq i \leq k-1$, denote the length of the corresponding shortest path. For such local paths, one can apply the abstract Barnette–Larman style bound of [16] to obtain much better worst-case bounds. Namely, we can show $K_i \leq 2^{O(n)} \log m$, $0 \leq i \leq k-1$, instead of $(2^{O(n)} \log m)^n$ (see Lemma 21). Crucially, the exact same independence and locality properties hold for these paths as for the shadow paths, due to the generality of our main locality lemma (Lemma 15). Furthermore, as these paths are only shorter than the corresponding shadow paths, their expected sum is again upper bounded by Borgwardt’s bound. With the improved worst-case bounds, our concentration estimates are sufficient to show that all paths indexed by planes in the net N have length $O(n^2 m^{\frac{1}{n-1}} + 2^{O(n)})$ with high probability.

1.3.2 The Lower Bound

For the lower bound, we first reduce to lower bounding the diameter of the polar polytope $P(A)^\circ = Q(A)$, where we show that $\text{diam}(P(A)) \geq (n-1)(\text{diam}(Q(A)) - 2)$ (see Lemma 23). This relation holds as long as $P(A)$ is a simple polytope containing the origin in its interior (which holds with probability $1 - 2^{-\Omega(m)}$). To prove it, we show that given any path between vertices v_1, v_2 of $P(A)$ of length D , respectively incident to distinct facets F_1, F_2 of $P(A)$, one can extract a facet path, where adjacent facets share an $n-2$ -dimensional intersection (i.e., a ridge), of length at most $D/(n-1) + 2$. Such facet paths exactly correspond to paths between vertices in $Q(A)$, yielding the desired lower bound.

For $m \geq 2^{\Omega(n)}$, proving that $\text{diam}(P(A)) \geq \Omega(nm^{1/(n-1)})$ reduces to showing that $\text{diam}(Q(A)) \geq m^{1/(n-1)}$ with high probability. For the $Q(A)$ lower bound, we examine the length of paths between vertices of $Q(A)$ maximizing antipodal objectives, e.g., $-e_1$ and e_1 . From here, one can one easily derive an $\Omega((m/\log m)^{\frac{1}{n-1}})$ lower bound on the length of such a path, by showing that every edge of $Q(A)$ has length $\epsilon := \Theta((\log m/m)^{\frac{1}{n-1}})$ and that the vertices in consideration are at distance $\Omega(1)$. This is a straightforward consequence of $Q(A)$ being tightly sandwiched by Euclidean balls, namely $(1 - \epsilon^2/2)\mathbb{B}_2^n \subseteq Q(A) \subseteq \mathbb{B}_2^n$ (Lemma 11) with high probability. This sandwiching property is itself a consequence of the rows of A being ϵ -dense on \mathbb{S}^{n-1} , as mentioned in the previous section.

Removing the extraneous logarithmic factor (which makes the multiplicative gap between our lower and upper bound go to infinity as $m \rightarrow \infty$), requires a much more involved argument as we cannot rely on a worst-case upper bound on the length of edges. Instead, we first associate any antipodal path above to a continuous curve on the sphere from $-e_1$ to e_1 (Lemma 26), corresponding to objectives maximized by vertices along the path. From here, we decompose any such curve into $\Omega(m^{\frac{1}{n-1}})$ segments whose endpoints are at distance $\Theta(m^{-1/(n-1)})$ on the sphere. Finally, by appropriately bucketing the breakpoints (Lemma 27) and applying a careful union bound, we show that for any such curve, an $\Omega(1)$ fraction of the segments induce at least 1 edge on the corresponding path with overwhelming probability (Theorem 28). For further details on the lower bound, including how we discretize the set of curves, we refer the reader to Section 4.

1.4 Organization

In Section 2 and the appendix, we introduce some basic notation as well as estimates on the measure of spherical caps. In Section 3, we prove the upper bound. Halfway into that section, we also prove Theorem 9, a tail bound on the shadow size that is of independent interest. We prove the lower bound in Section 4. Most proofs are organized in the different sections of the appendix. Any proof not present in the text can be found in the appendix.

2 Preliminaries

For notational simplicity in the sequel, it will be convenient to treat A as a subset of \mathbb{S}^{n-1} instead of a matrix. For $A \subseteq \mathbb{S}^{n-1}$, we will slightly abuse notation and let $P(A) := \{x \in \mathbb{R}^n : \langle x, a \rangle \leq 1, \forall a \in A\}$ and $Q(A) := \text{conv}(A)$. We denote the indicator of a random event X by $1[X]$, i.e., $1[X] = 1$ if X and $1[X] = 0$ otherwise. For $k \in \mathbb{N}$ we write $[k] := \{1, \dots, k\}$.

Our objects of interest are defined as follows:

► **Definition 2.** For any polyhedron $P \subseteq \mathbb{R}^n$, a path is a sequence $v_1, v_2, \dots, v_k \in P$ of vertices, such that each line segment $[v_i, v_{i+1}]$, $i \in [k-1]$, is an edge of P . A path is monotone with respect to an inner product $\langle w, \cdot \rangle$ if $\langle w, v_{i+1} \rangle \geq \langle w, v_i \rangle$ for every $i \in [k-1]$.

The distance between vertices $v_1, v_2 \in P$ is the minimum number k such that there exists a path $v'_1, v'_2, \dots, v'_{k+1}$ with $v_1 = v'_1$ and $v'_k = v_2$. The diameter of P is the maximal distance between any two of its vertices.

2.1 Density Estimates

In this section, we give bounds on the fineness of the net induced by a Poisson distributed subset of \mathbb{S}^{n-1} . Roughly speaking, if the set A follows a $\text{Pois}(\mathbb{S}^{n-1}, m)$ distribution then A will be $\Theta((\log m/m)^{1/(n-1)})$ -dense, see Definition 3. While this estimate is standard in the stochastic geometry, it is not so easy to find a reference giving quantitative probabilistic bounds, as more attention has been given to establishing exact asymptotics as $m \rightarrow \infty$ (see [27]). We provide a simple proof of this fact here, together with the probabilistic estimates that we will need.

► **Definition 3.** For $w \in \mathbb{S}^{n-1}$ and $r \geq 0$, we denote by $C(w, r) = \{x \in \mathbb{S}^{n-1} : \|w - x\| \leq r\}$ the spherical cap of radius r centered at w .

We say $A \subseteq \mathbb{S}^{n-1}$ is ε -dense in the sphere for $\varepsilon > 0$ if for every $w \in \mathbb{S}^{n-1}$ there exists $a \in A$ such that $a \in C(w, \varepsilon)$.

► **Lemma 4.** For $m \geq n \geq 2$ and $0 < p < m^{-n}$, have $\varepsilon = \varepsilon(m, n, p) > 0$ satisfy $\sigma(C(v, \varepsilon)) = 3e \log(1/p)/m < 1/12$. Then, for $A \sim \text{Pois}(\mathbb{S}^{n-1}, m)$,

$$\Pr[\exists v \in \mathbb{S}^{n-1} : C(v, \varepsilon) \cap A = \emptyset] \leq p$$

and for every $t \geq 1$,

$$\Pr[\exists v \in \mathbb{S}^{n-1} : |C(v, t\varepsilon) \cap A| \geq 45 \log(1/p)t^{n-1}] \leq p.$$

We now give effective bounds on the density estimate ε above. Note that taking the $(n-1)^{\text{th}}$ root of the bounds for ε^{n-1} below yields $\varepsilon = \Theta((\log m/m)^{1/(n-1)})$ for $m = n^{\Omega(1)}$ and $p = 1/m^{-n}$.

► **Corollary 5.** Let $\varepsilon > 0$ be as in Lemma 4, i.e., satisfying $\sigma(C(v, \varepsilon)) = 3e \log(1/p)/m \leq 1/12$. Then $\varepsilon \in [0, \sqrt{2(1 - \frac{2}{\sqrt{n}})}]$ and

$$12e \log(1/p)/m \leq \varepsilon^{n-1} \leq (\sqrt{2})^{n-1} \cdot 18\sqrt{n} \log(1/p)/m.$$

3 Shadow size and upper bounding the diameter

In the first part of this section, we prove a concentration result on the number of *shadow vertices* of $P(A)$. This addresses an open problem from [6]. In the second part, we use the resulting tools to prove Theorem 10, our high-probability upper bound on the diameter of $P(A)$. We start by defining a useful set of paths for which we know their expected lengths.

► **Definition 6.** Let $P \subseteq \mathbb{R}^n$ be a polyhedron and $W \subseteq \mathbb{R}^n$ be a two-dimensional linear subspace. We denote by $\mathcal{S}(P, W)$ the set of shadow vertices: the vertices of P that maximize a non-zero objective function $\langle w, \cdot \rangle$ with $w \in W$.

From standard polyhedral theory, we get a characterization of shadow vertices:

► **Lemma 7.** Let $P(A)$ be a polyhedron given by $A \subseteq \mathbb{R}^n$ and $w \in \mathbb{R}^n \setminus \{0\}$. A vertex $v \in P(A)$ maximizes $\langle w, \cdot \rangle$ if and only if $w\mathbb{R}_+ \cap \text{conv}\{a \in A : \langle a, v \rangle = 1\} \neq \emptyset$.

Hence for $W \subseteq \mathbb{R}^n$ a two-dimensional linear subspace, a vertex $v \in P(A)$ is a shadow vertex $v \in \mathcal{S}(P(A), W)$ if and only if $\text{conv}\{a \in A : \langle a, v \rangle = 1\} \cap W \setminus \{0\} \neq \emptyset$.

The set of shadow vertices for a fixed plane W induces a connected subgraph in the graph consisting of vertices and edges of P , and so any two shadow vertices are connected by a path of length at most $|\mathcal{S}(P, W)|$. As such, for nonzero $w_1, w_2 \in W$, we might speak of a *shadow path* from w_1 to w_2 to denote a path from a maximizer of $\langle w_1, \cdot \rangle$ to a maximizer of $\langle w_2, \cdot \rangle$ that stays inside $\mathcal{S}(P, W)$ and is monotonous with respect to $\langle w_2, \cdot \rangle$. The shadow path was studied by Borgwardt:

► **Theorem 8** ([6, 7]). Let $m \geq n$ and fix a two-dimensional linear subspace $W \subseteq \mathbb{R}^n$. Pick any probability distribution on \mathbb{R}^n that is invariant under rotations and let the entries of $A \subseteq \mathbb{R}^n$, $|A| = m$, be independently sampled from this distribution. Then, almost surely, for any linearly independent $w_1, w_2 \in W$ there is a unique shadow path from w_1 to w_2 . Moreover, the vertices in $\mathcal{S}(P(A), W)$ are in one-to-one correspondence to the vertices of $\pi_W(P(A))$, the orthogonal projection of $P(A)$ onto W . The expected length of the shadow path from w_1 to w_2 is at most

$$\mathbb{E}[|\mathcal{S}(P(A), W)|] = O(n^2 m^{\frac{1}{n-1}}).$$

This upper bound is tight up to constant factors for the uniform distribution on \mathbb{S}^{n-1} .

We prove a tail bound for the shadow size when $A \sim \text{Pois}(\mathbb{S}^{n-1}, m)$. This result answers a question of Borgwardt in the asymptotic regime, regarding whether bounds on higher moments of the shadow size can be given. To obtain such concentration, we show that the shadow decomposes into a sum of nearly independent “local shadows”, using that A will be ε -dense per Lemma 4, allowing us to apply standard concentration results for sums of nearly independent random variables.

► **Theorem 9** (Shadow Size Concentration). *Let $e^{\frac{-m}{18\sqrt{n}(76\sqrt{2})^{n-1}}} < p < m^{-2n}$ and let*

$$t_p := \max \left(\sqrt{O(Un^2 m^{\frac{1}{n-1}} \log(1/p))}, O(U \log(1/p)) \right)$$

for $U := O(n2^{n^2} (\log(1/p))^n)$. *If $A \sim \text{Pois}(\mathbb{S}^{n-1}, m)$ then the shadow size satisfies*

$$\Pr \left[\left| |\mathcal{S}(P(A), W)| - \mathbb{E}[|\mathcal{S}(P(A), W)|] \right| > t_p \right] \leq 4p.$$

In the second part of this section, we extend the resulting tools to obtain our upper bound on the diameter.

► **Theorem 10** (Diameter Upper Bound). *Assume that p satisfies $e^{\frac{-m}{18\sqrt{n}(76\sqrt{2})^{n-1}}} < p < m^{-2n}$. If $A = \{a_1, \dots, a_M\} \in \mathbb{S}^{n-1}$, where M is Poisson with $\mathbb{E}[M] = m$, and a_1, \dots, a_M are uniformly and independently distributed in \mathbb{S}^{n-1} . Then, we have that*

$$\Pr[\text{diam}(P(A)) > O(n^2 m^{\frac{1}{n-1}} + n4^n \log^2(1/p))] \leq O(\sqrt{p}).$$

3.1 Only “nearby” constraints are relevant

We will start by showing that, with very high probability, constraints that are “far away” from a given point on the sphere will not have any impact on the local shape of paths. That will result in a degree of independence between different parts of the sphere, which will be essential in getting concentration bounds on key quantities.

► **Lemma 11.** *If $A \subseteq \mathbb{S}^{n-1}$ is ε -dense for $\varepsilon \in [0, \sqrt{2})$ then $\mathbb{B}_2^n \subseteq P(A) \subseteq \left(1 - \frac{\varepsilon^2}{2}\right)^{-1} \mathbb{B}_2^n$.*

► **Lemma 12.** *If $w \in \mathbb{S}^{n-1}$, $\alpha < 1$, $\|v\| \leq (1 - \alpha)^{-1}$ and $\langle v, w \rangle \geq 1$ then $\|v/\|v\| - w\|^2 \leq 2\alpha$.*

Proof. We have $1 \leq \langle v, w \rangle = \|v\| \cdot \langle v/\|v\|, w \rangle \leq (1 - \alpha)^{-1} \langle v/\|v\|, w \rangle$. Hence $1 - \|v/\|v\| - w\|^2/2 = \langle v/\|v\|, w \rangle \geq 1 - \alpha$, which exactly implies that $\|v/\|v\| - w\|^2 \leq 2\alpha$ as required. ◀

We will use the above lemmas to prove the main technical estimate of this subsection: if $A \subseteq \mathbb{S}^{n-1}$ is ε -dense and $w_1, w_2 \in \mathbb{S}^{n-1}$ satisfy $\|w_1 - w_2\| \leq 2\varepsilon/n$ then any vertex on any path on $P(A)$ starting at a maximizer of $\langle w_1, \cdot \rangle$ that is non-decreasing with respect to $\langle w_2, \cdot \rangle$ can only be tight at constraints $\langle a, x \rangle = 1$ induced by $a \in A \cap C(w_2, (2 + 2/n)\varepsilon)$. All other constraints are strictly satisfied by every vertex on such a monotone path.

► **Lemma 13.** *Let $\varepsilon \in [0, 1]$ and assume that $w_1, w_2 \in \mathbb{S}^{n-1}$ satisfy $\|w_1 - w_2\| \leq (1 - \varepsilon^2/2)$. Let $v_1, v \in \mathbb{R}^n$ satisfy $\langle w_1, v_1 \rangle \geq 1$ and $\langle w_2, v \rangle \geq \langle w_2, v_1 \rangle$, and assume $\|v_1\|, \|v\| \leq (1 - \varepsilon^2/2)^{-1}$. Last, let $a \in \mathbb{S}^{n-1}$ satisfy $\langle a, v \rangle \geq 1$. Then we have $\|w_2 - a\| \leq 2\varepsilon + \|w_1 - w_2\|$.*

To round out this subsection, we prove that the conclusion of Lemma 13 holds whenever $v, v_1 \in P(A)$ and A is ε -dense in a neighbourhood around w_2 .

► **Definition 14.** *Given sets $A, C \subseteq \mathbb{S}^{n-1}$ and $\varepsilon > 0$, we say that A is ε -dense for C if for every $c \in C$ there exists $a \in A$ such that $\|a - c\| \leq \varepsilon$.*

18:10 Asymptotic Bounds on the Combinatorial Diameter of Random Polytopes

► **Lemma 15.** *Let $A \subseteq \mathbb{S}^{n-1}$ be compact and ε -dense for $C(w_2, 4\varepsilon)$, $\varepsilon > 0$. Let $v_1, v \in P(A)$ and $w_1, w_2 \in \mathbb{S}^{n-1}$ satisfying $\langle w_1, v_1 \rangle \geq 1$, $\langle w_2, v \rangle \geq \langle w_2, v_1 \rangle$ and $\|w_1 - w_2\| \leq \varepsilon$. Now let $a \in \mathbb{S}^{n-1}$ satisfy $\langle a, v \rangle \geq 1$. Then we have $\|v_1\|, \|v\| \leq (1 - \varepsilon^2/2)^{-1}$ and $\|w_2 - a\| \leq 2\varepsilon + \|w_1 - w_2\|$.*

Note also the contrapositive of the above statement: for w_1, w_2, v_1, v, A satisfying the conditions above, we have for $a \in \mathbb{S}^{n-1}$ that $\|w_2 - a\| > 2\varepsilon + \|w_1 - w_2\|$ implies $\langle a, v \rangle < 1$.

3.2 Locality, independence, and concentration

With an eye to Lemma 15, this subsection is concerned with proving concentration for sums of random variables that behave nicely when A is dense in given neighbourhoods. The specific random variables that we will use this for are the paths between the maximizers of nearby objective vectors $w_1, w_2 \in \mathbb{S}^{n-1}$.

► **Definition 16.** *Given m, n, p , let $\varepsilon = \varepsilon(m, n, p) > 0$ be as in Lemma 4 and $A \subseteq \mathbb{R}^n$ be a random finite set. For $x, y \in \mathbb{S}^{n-1}$ define the event $E_{x,y}$ as:*

- *A is ε -dense for $C(x, \|x - y\| + 4\varepsilon)$, and*
- *for every $z \in [x, y]$ we have*

$$\left| A \cap C\left(\frac{z}{\|z\|}, (2 + 2/n)\varepsilon\right) \right| \leq 45e2^n \log(1/p)$$

A random variable K is called (x, y) -local if $E_{x,y}$ implies that K is a function of $A \cap C(x, 5\varepsilon + \|x - y\|)$.

In particular, we will use that if K is (x, y) -local then $K1[E_{x,y}]$ is a function of $A \cap C(x, 5\varepsilon + \|x - y\|)$.

To help prove that certain paths are local random variables, we will use the following helper lemma.

► **Lemma 17.** *Let $w_1, w_2 \in \mathbb{S}^{n-1}$, and have $w_1 = v_1, v_2, \dots, v_{k+1} = w_2$ be equally spaced on a shortest geodesic segment on \mathbb{S}^{n-1} connecting w_1 and w_2 . Then for every $i \in [k]$ we have $\|w_1 - w_2\|/k \leq \|v_i - v_{i+1}\| \leq \pi\|w_1 - w_2\|/k$.*

Many paths on $P(A)$ turn out to be such local random variables. One example are short segments of the shadow paths from Theorem 8.

► **Lemma 18.** *Let $w_1, w_2 \in \mathbb{S}^{n-1}$ satisfy $\|w_1 - w_2\| \leq \varepsilon$. Then the length of the shadow path on $P(A)$ from w_1 to w_2 is a (w_1, w_2) -local random variable. If $\|w_1 - w_2\| \leq \varepsilon$ then E_{w_1, w_2} implies that this path has length at most $2n(45e2^n \log(1/p))^n$.*

► **Lemma 19.** *Let $0 < p < m^{-2n}$ and let $\varepsilon = \varepsilon(m, n, p) < 1/76$ be as in Lemma 4 and let $k \geq 2\pi/\varepsilon$ be the smallest number divisible by 76. Let $W \subseteq \mathbb{R}^n$ be a fixed 2D linear subspace and let $w_1, \dots, w_k, w_{k+1} = w_1 \in W \cap \mathbb{S}^{n-1}$ be equally spaced around the circle. Assume for every $i \in [k]$ that $K_i \geq 0$ is a (w_i, w_{i+1}) -local random variable and there exists $U \leq m^n$ such that $K_i \leq U$ whenever $E_{w_i, w_{i+1}}$. Furthermore assume that $\mathbb{E}[\sum_{i=1}^k K_i] \leq O(n^2 m^{\frac{1}{n-1}})$. Then*

$$\Pr \left[\left| \sum_{i \in [k]} K_i - \mathbb{E} \left[\sum_{i \in [k]} K_i \right] \right| \geq t_p \right] \leq 4p$$

for $t_p = \max \left(\sqrt{O(n^2 m^{\frac{1}{n-1}} \log(1/p))}, O(U \log(1/p)) \right)$.

3.3 Upper bound on the diameter

In this section we prove our high probability upper bound on $\text{diam}(P(A))$. We start by proving that for fixed W the vertices in $\mathcal{S}(P(A), W)$ are connected by short paths, where we aim for an error term smaller than that of Theorem 9. We require the following abstract diameter bound from [16]. We will only need the Barnette–Larman style bound.

► **Theorem 20.** *Let $G = (V, E)$ be a connected graph, where the vertices V of G are subsets of $\{1, \dots, k\}$ of cardinality n and the edges E of G are such that for each $u, v \in V$ there exists a path connecting u and v whose intermediate vertices all contain $u \cap v$.*

Then the following upper bounds on the diameter of G hold:

$$2^{n-1} \cdot k - 1 \text{ (Barnette–Larman)}, \quad k^{1+\log n} - 1 \text{ (Kalai–Kleitman)}.$$

To confirm that the above theorem indeed gives variants of the Barnette–Larman and Kalai–Kleitman bounds, let $A = \{a_1, \dots, a_m\} \subseteq \mathbb{S}^{n-1}$ be in general position. For a vertex $x \in P(A)$, we denote $A_x = \{a \in A : \langle a, x \rangle = 1\}$. Consider the following sets

$$V = \{A_x : x \text{ is a vertex of } P(A)\}, \\ E = \{\{A_x, A_y\} : [x, y] \text{ is an edge of } P(A)\}.$$

One can check that $G = (V, E)$ satisfies almost surely the assumptions of Theorem 20 which therefore shows that the combinatorial diameter of $P(A)$ is less than $\min(2^{n-1} \cdot m - 1, m^{1+\log n} - 1)$. Up to a constant factor difference, these bounds correspond to the same bounds described in the introduction.

Now we use the Barnette–Larman style bound to bound the length of the local paths.

► **Lemma 21.** *Let $w_1, w_2 \in \mathbb{S}^{n-1}$ satisfy $\|w_1 - w_2\| \leq \varepsilon$, where $\varepsilon = \varepsilon(m, n, p)$ is as in Lemma 4. Furthermore, let K denote the maximum over all $w \in [w_1, w_2]$ of the length of the shortest path from a maximizer $v_w \in P(A)$ of $\langle w, \cdot \rangle$ to the maximizer of $\langle w_2, \cdot \rangle$ of which every vertex $v \in P(A)$ on the path satisfies $\langle w_2, v \rangle \geq \langle w_2, v_w \rangle$. Then K is a (w_1, w_2) -local random variable and E_{w_1, w_2} implies that K_i is at most $45en4^n \log(1/p)$.*

► **Theorem 22.** *Let $0 < p < m^{-2n}$ and let*

$$t_p = \max \left(\sqrt{O(Un^2 m^{\frac{1}{n-1}} \log(1/p))}, O(U \log(1/p)) \right)$$

for $U = O(n4^n \log(1/p))$. If $W \subseteq \mathbb{R}^n$ is a fixed 2D linear subspace and $A \sim \text{Pois}(\mathbb{S}^{n-1}, m)$, the largest distance T between any two shadow vertices satisfies

$$\Pr[T \geq O(n^2 m^{\frac{1}{n-1}}) + t_p] \leq 4p$$

Proof. Let w_1, \dots, w_k be as in Lemma 19 and let K_i denote the maximum over all $w \in [w_i, w_{i+1}]$ of the length of the shortest path from a shadow vertex v_w maximizing $\langle w, \cdot \rangle$ to a vertex maximizing $\langle w_{i+1}, \cdot \rangle$ such that every vertex v on this path satisfies $\langle w_{i+1}, v \rangle \geq \langle w_{i+1}, v_w \rangle$. From Lemma 21 we know that K_i is a (w_i, w_{i+1}) -local random variable and $K_i \leq 45en4^n \log(1/p)$ whenever $E_{w_i, w_{i+1}}$. Now recall Theorem 8. Observe that $T \leq \sum_{i \in [k]} K_i$ almost surely by concatenating the above-mentioned paths, and note that that $\sum_{i \in [k]} K_i \leq \mathcal{S}(P(A), W)$ holds almost surely, which implies $\mathbb{E}[\sum_{i \in [k]} K_i] = O(n^2 m^{\frac{1}{n-1}})$. We apply Lemma 19 to $\sum_{i \in [k]} K_i$ and get the desired result. ◀

► **Theorem 10** (Diameter Upper Bound). Assume that p satisfies $e^{\frac{-m}{18\sqrt{n}(76\sqrt{2})^{n-1}}} < p < m^{-2n}$. If $A = \{a_1, \dots, a_M\} \in \mathbb{S}^{n-1}$, where M is Poisson with $\mathbb{E}[M] = m$, and a_1, \dots, a_M are uniformly and independently distributed in \mathbb{S}^{n-1} . Then, we have that

$$\Pr[\text{diam}(P(A)) > O(n^2 m^{\frac{1}{n-1}} + n4^n \log^2(1/p))] \leq O(\sqrt{p}).$$

4 Lower Bounding the Diameter of $P(A)$

To begin, we first reduce to lower bounding the diameter of the polar polytope P° , corresponding to a convex hull of m uniform points on \mathbb{S}^{n-1} , via the following simple lemma.

► **Lemma 23** (Diameter Relation). For $n \geq 2$, let $P \subseteq \mathbb{R}^n$ be a simple bounded polytope containing the origin in its interior and let $Q = P^\circ := \{x \in \mathbb{R}^n : \langle x, y \rangle \leq 1, \forall y \in P\}$ denote the polar of P . Then, $\text{diam}(P) \geq (n-1)(\text{diam}(Q) - 2)$.

We then associate any “antipodal” path to a continuous curve on the sphere corresponding to objectives maximized by vertices along the path. From here, we decompose any such curve into $\Omega(m^{\frac{1}{n-1}})$ segments whose endpoints are at distance $\Theta(m^{-1/(n-1)})$ on the sphere. Finally, we apply a suitable union bound, to show that for any such curve, an $\Omega(1)$ fraction of the segments induce at least 1 edge on the corresponding path.

Building on Lemma 23, we turn to proving the lower bound for $Q(A)$.

For a discrete set $N \subseteq \mathbb{S}^{n-1}$, a point $x_0 \in N$ and a positive number $\varepsilon > 0$ we denote by

$$X_k := X_k(N, x_0, \varepsilon) = \{\mathbf{x} \in N^k : x_i \neq x_j \text{ and } 6\varepsilon \leq \|x_i - x_{i+1}\| \leq 8\varepsilon \text{ for any } 0 \leq i < j \leq k\}$$

the set of all sequences of k distinct points in N with jumps of length between 6ε and 8ε (including an extra initial jump between x_0 and x_1).

► **Lemma 24.** Let $\varepsilon > 0$. If $N \subseteq \mathbb{S}^{n-1}$ is a maximal ε -separated set, then

$$|X_k| \leq (17^{n-1})^k.$$

Note that a maximal ε -separated set is also an ε -net.

► **Lemma 25.** Let $f: [0, 1] \rightarrow \mathbb{S}^{n-1}$ be a continuous function. Let $\varepsilon > 0$ and $N \subseteq \mathbb{S}^{n-1}$ be an ε -net, such that $f(0) \in N$. There exist $k \in \mathbb{N}_0$, $0 \leq t_0 < t_1 < \dots < t_k \leq 1$ and $x_0, \dots, x_k \in N$ such that

1. $\|f(t_i) - x_i\| \leq \varepsilon$ for any $i \in \{0, \dots, k\}$,
2. $\|f(t) - x_i\| \geq \varepsilon$ for any $i \in \{0, \dots, k\}$ and $t > t_i$,
3. $(x_1, \dots, x_k) \in X_k(N, x_0, \varepsilon)$,
4. $\|x_k - f(1)\| < 7\varepsilon$.

► **Lemma 26.** Let $A \subseteq \mathbb{S}^{n-1}$ be a finite subset of the sphere. Let $[a_0, a_1], [a_1, a_2], \dots, [a_{\ell-1}, a_\ell]$ be a path along the edges of $Q(A)$. There exists a continuous function $f: [0, 1] \rightarrow \mathbb{S}^{n-1}$ and $0 = s_0 < s_1 < \dots < s_{\ell+1} = 1$ such that $f(0) = a_0$, $f(1) = a_\ell$, and for any $i \in \{0, 1, \dots, \ell\}$ and any $t \in [s_i, s_{i+1}]$,

$$a_i \in \operatorname{argmin}_{a \in A} (\|f(t) - a\|).$$

► **Lemma 27.** Let $A \subseteq \mathbb{S}^{n-1}$ be a finite subset of the sphere, containing two points $a_+, a_- \in A$ such that $\|a_+ - a_-\| \geq 1$. Let $\varepsilon > 0$ and $N \subseteq \mathbb{S}^{n-1}$ be a maximal ε -separated set, such that $a_+ \in N$. Set $x_0 = a_+$ and $k_0 = \lceil 1/8\varepsilon \rceil - 1$. It holds that

$$\text{diam}(Q(A)) \geq \min_{k \geq k_0} \min_{\mathbf{x} \in X_k(N, x_0, \varepsilon)} \sum_{0 \leq i \leq k-1} 1[C(x_i, \varepsilon/2) \cap A \neq \emptyset] 1[C(x_{i+1}, \varepsilon/2) \cap A \neq \emptyset].$$

► **Theorem 28** (Lower Bound for $Q(A)$). *There exist positive constants $c_2 < 1$ and $c_3 > 1$ independent of $n \geq 3$ and m such that the following holds. Let $A = \{a_1, \dots, a_M\} \in \mathbb{S}^{n-1}$, where M is Poisson with $\mathbb{E}[M] = m$, and a_1, \dots, a_M are uniformly and independently distributed in \mathbb{S}^{n-1} . Then, with probability at least $1 - e^{-c_3^{n-1} m^{1/(n-1)}}$, the combinatorial diameter of $Q(A)$ is at least $c_2 m^{1/(n-1)}$.*

Proof. Without loss of generality $m \geq (1/c_2)^{n-1}$ since otherwise the statement of the theorem is trivial.

In this proof the constants $1 < c_3 < c_4 < c_5 < c_6 < c_2^{-1}$ are large enough constants, independent from n and m .

We set $\varepsilon = c_6 m^{-1/(n-1)}$, and want to apply Lemma 27. Let N be an ε -net, obtained from a maximal ε -separated set, such that it contains a point a_+ from the set A . For independence properties needed later we take a_+ randomly and uniformly from the set A . With probability $1 - e^{-m/2}$ we have that A intersects the halfsphere $\{u \in \mathbb{S}^{n-1} : \langle a_+, u \rangle \leq 0\}$. In which case there exists a point $a_- \in A$ such that $\|a_+ - a_-\| \geq \sqrt{2} \geq 1$. Therefore we can apply Lemma 27 with $x_0 = a_+$. Combined with the union bound, we get

$$\Pr \left(\text{diam } Q(A) \leq c_2 m^{1/(n-1)} \right) \leq e^{-m/2} + \sum_{\substack{k \geq k_0 \\ \mathbf{x} \in X_k(N, x_0, \varepsilon)}} \Pr \left(\sum_{0 \leq i \leq k-1} B_i \leq c_2 m^{1/(n-1)} \right),$$

where

$$k_0 = \lceil 1/8\varepsilon \rceil + 1 \geq 1/8\varepsilon = m^{1/(n-1)}/8c_6,$$

and the summands in the probability are Bernoulli random variables

$$B_i = 1[C(x_i, \varepsilon/2) \cap A \neq \emptyset]1[C(x_{i+1}, \varepsilon/2) \cap A \neq \emptyset].$$

For $1 \leq i \leq k-1$, they are identically distributed, with failure probability

$$\begin{aligned} \Pr(B_i = 0) &\leq 2\Pr(C(x_i, \varepsilon/2) \cap A = \emptyset) = 2\exp(-m\sigma(C(x_i, \varepsilon/2))) \\ &\leq 2\exp(-m(\varepsilon/4)^{n-1}) = 2\exp\left(-\left(\frac{c_6}{4}\right)^{n-1}\right) =: 1 - p. \end{aligned}$$

Note that we lower bounded the volume of the cap $\sigma(C(x_i, \varepsilon/2)) \geq (\varepsilon/4)^{n-1}\sigma(C(x_i, 2))$. Since N forms a maximal ε -separated set and the x_i are distinct, the caps $C(x_i, \varepsilon/2)$ are disjoint and therefore the random variables B_1, B_3, B_5, \dots are independent. Next we exploit this independence. Let $k \geq k_0$, and set $K = \lfloor k/2 \rfloor$. Note that $K \geq 1/16\varepsilon = m^{1/(n-1)}/16c_6$. Assuming that $c_2 \leq 1/32c_6$, we have

$$\Pr \left(\sum_{0 \leq i \leq k-1} B_i \leq c_2 m^{1/(n-1)} \right) \leq \Pr \left(\sum_{1 \leq i \leq K} B_{2i-1} \leq \frac{K}{2} \right) = \sum_{1 \leq i \leq \lfloor K/2 \rfloor} \binom{K}{i} p^i (1-p)^{K-i}.$$

Now we bound p by 1, $(1-p)^{K-i}$ by $(1-p)^{K/2}$ and $\sum \binom{K}{i}$ by 2^K , which provides us the bound

$$\Pr \left(\sum_{0 \leq i \leq k-1} B_i \leq c_2 m^{-1/(n-1)} \right) \leq (2(1-p)^{1/2})^K \leq \left(e^{(-c_5^{n-1})} \right)^K.$$

Thus, with the bound $|X_k| \leq (17^{n-1})^k$ from lemma 24, and the fact that $K \geq k/2$, we get

$$\begin{aligned} \Pr\left(\text{diam } Q(A) \leq c_2 m^{-1/(n-1)}\right) &\leq e^{-m/2} + \sum_{k \geq k_0} \left(e^{(-\frac{1}{2}(c_5)^{n-1} + (n-1) \ln 17)}\right)^k \\ &\leq e^{-m/2} + \sum_{k \geq k_0} (e^{-(c_4)^{n-1}})^k \\ &= e^{-m/2} + \frac{e^{-k_0 c_4^{n-1}}}{1 - e^{-(c_4)^{n-1}}} \\ &\leq e^{-m/2} + \frac{e^{-\frac{m^{1/(n-1)}}{8c_6} c_4^{n-1}}}{1 - e^{-c_4^{n-1}}} \\ &\leq e^{-c_3^{n-1} m^{1/(n-1)}}. \quad \blacktriangleleft \end{aligned}$$

References

- 1 Michel L Balinski. The Hirsch conjecture for dual transportation polyhedra. *Mathematics of Operations Research*, 9(4):629–633, 1984.
- 2 Imre Bárány and Zoltán Füredi. On the shape of the convex hull of random points. *Probability Theory and Related Fields*, 77(2):231–240, February 1988. doi:10.1007/bf00334039.
- 3 David Barnette. Wv paths on 3-polytopes. *Journal of Combinatorial Theory*, 7(1):62–70, July 1969. doi:10.1016/s0021-9800(69)80007-4.
- 4 David Barnette. An upper bound for the diameter of a polytope. *Discrete Mathematics*, 10(1):9–13, 1974. doi:10.1016/0012-365x(74)90016-8.
- 5 Nicolas Bonifas, Marco Di Summa, Friedrich Eisenbrand, Nicolai Hähnle, and Martin Niemeier. On sub-determinants and the diameter of polyhedra. *Discrete & Computational Geometry*, 52(1):102–115, 2014.
- 6 Karl Heinz Borgwardt. *The simplex method: a probabilistic analysis*, volume 1 of *Algorithms and Combinatorics: Study and Research Texts*. Springer-Verlag, Berlin, 1987. doi:10.1007/978-3-642-61578-8.
- 7 Karl Heinz Borgwardt. Erratum: A sharp upper bound for the expected number of shadow vertices in lp-polyhedra under orthogonal projection on two-dimensional planes. *Mathematics of Operations Research*, 24(4):925–984, 1999. URL: <http://www.jstor.org/stable/3690611>.
- 8 Karl Heinz Borgwardt and Petra Huhn. A lower bound on the average number of pivot-steps for solving linear programs valid for all variants of the simplex-algorithm. *Mathematical Methods of Operations Research*, 49(2):175–210, April 1999. doi:10.1007/s186-1999-8373-5.
- 9 Steffen Borgwardt, Jesús A De Loera, and Elisabeth Finhold. The diameters of network-flow polytopes satisfy the Hirsch conjecture. *Mathematical Programming*, 171(1):283–309, 2018.
- 10 Graham Brightwell, Jan Van den Heuvel, and Leen Stougie. A linear bound on the diameter of the transportation polytope. *Combinatorica*, 26(2):133–139, 2006.
- 11 Daniel Dadush and Nicolai Hähnle. On the shadow simplex method for curved polyhedra. *Discrete Computational Geometry*, 56(4):882–909, June 2016. doi:10.1007/s00454-016-9793-3.
- 12 Daniel Dadush and Sophie Huiberts. A friendly smoothed analysis of the simplex method. *SIAM Journal on Computing*, 49(5):STOC18–449, 2019.
- 13 Alberto Del Pia and Carla Michini. On the diameter of lattice polytopes. *Discrete & Computational Geometry*, 55(3):681–687, 2016.
- 14 Antoine Deza and Lionel Pournin. Improved bounds on the diameter of lattice polytopes. *Acta Mathematica Hungarica*, 154(2):457–469, 2018.
- 15 Martin Dyer and Alan Frieze. Random walks, totally unimodular matrices, and a randomised dual simplex algorithm. *Mathematical Programming*, 64(1):1–16, 1994.
- 16 Friedrich Eisenbrand, Nicolai Hähnle, Alexander Razborov, and Thomas Rothvoss. Diameter of polyhedra: Limits of abstraction. *Mathematics of Operations Research*, 35(4):786–794, 2010.

- 17 Marc Glisse, Sylvain Lazard, Julien Michel, and Marc Pouget. Silhouette of a random polytope. *Journal of Computational Geometry*, 7(1):14, 2016.
- 18 Richard C Grinold. The Hirsch conjecture in Leontief substitution systems. *SIAM Journal on Applied Mathematics*, 21(3):483–485, 1971.
- 19 Gil Kalai and Daniel J. Kleitman. A quasi-polynomial bound for the diameter of graphs of polyhedra. *Bull. Amer. Math. Soc.*, 26(2):315–317, July 1992. doi:10.1090/s0273-0979-1992-00285-9.
- 20 Victor Klee, David W Walkup, et al. The d -step conjecture for polyhedra of dimension $d < 6$. *Acta Mathematica*, 117:53–78, 1967.
- 21 Peter Kleinschmidt and Shmuel Onn. On the diameter of convex polytopes. *Discrete mathematics*, 102(1):75–77, 1992.
- 22 Jean-Philippe Labbé, Thibault Manneville, and Francisco Santos. Hirsch polytopes with exponentially long combinatorial segments. *Mathematical Programming*, 165(2):663–688, 2017.
- 23 D.G. Larman. Paths on polytopes. *Proc. London Math. Soc. (3)*, s3-20(1):161–178, January 1970. doi:10.1112/plms/s3-20.1.161.
- 24 Carla Michini and Antonio Sassano. The Hirsch Conjecture for the fractional stable set polytope. *Mathematical Programming*, 147(1):309–330, 2014.
- 25 Denis Naddef. The Hirsch conjecture is true for $(0, 1)$ -polytopes. *Mathematical Programming: Series A and B*, 45(1-3):109–110, 1989.
- 26 Hariharan Narayanan, Rikhav Shah, and Nikhil Srivastava. A spectral approach to polytope diameter, 2021. arXiv:2101.12198.
- 27 A Reznikov and EB Saff. The covering radius of randomly distributed points on a manifold. *International Mathematics Research Notices*, 2016(19):6065–6094, 2016.
- 28 Laura Sanità. The diameter of the fractional matching polytope and its hardness implications. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 910–921. IEEE, 2018.
- 29 Francisco Santos. A counterexample to the Hirsch Conjecture. *Annals of Mathematics*, 176(1):383–412, July 2012. doi:10.4007/annals.2012.176.1.7.
- 30 Rolf Schneider and Wolfgang Weil. *Stochastic and integral geometry*. Probability and its Applications (New York). Springer-Verlag, Berlin, 2008. doi:10.1007/978-3-540-78859-1.
- 31 Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.
- 32 Noriyoshi Sukegawa. An asymptotically improved upper bound on the diameter of polyhedra. *Discrete & Computational Geometry*, 62(3):690–699, 2019.
- 33 Michael J Todd. An improved Kalai–Kleitman bound for the diameter of a polyhedron. *SIAM Journal on Discrete Mathematics*, 28(4):1944–1947, 2014.
- 34 Roman Vershynin. Beyond Hirsch conjecture: walks on random polytopes and smoothed complexity of the simplex method. *SIAM J. Comput.*, 39(2):646–678, 2009. Preliminary version in FOCS ‘06. doi:10.1137/070683386.

Signed Barcodes for Multi-Parameter Persistence via Rank Decompositions

Magnus Bakke Botnan ✉

Vrije Universiteit Amsterdam, The Netherlands

Steffen Oppermann ✉

Norwegian University of Science and Technology, Trondheim, Norway

Steve Oudot ✉

Inria, Palaiseau, France

Abstract

In this paper we introduce the signed barcode, a new visual representation of the global structure of the rank invariant of a multi-parameter persistence module or, more generally, of a poset representation. Like its unsigned counterpart in one-parameter persistence, the signed barcode encodes the rank invariant as a \mathbb{Z} -linear combination of rank invariants of indicator modules supported on segments in the poset. It can also be enriched to encode the generalized rank invariant as a \mathbb{Z} -linear combination of generalized rank invariants in fixed classes of interval modules. In the paper we develop the theory behind these rank decompositions, showing under what conditions they exist and are unique – so the signed barcode is canonically defined. We also illustrate the contribution of the signed barcode to the exploration of multi-parameter persistence modules through a practical example.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology

Keywords and phrases Topological data analysis, multi-parameter persistent homology

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.19

Related Version *Full Version*: <https://arxiv.org/abs/2107.06800>

1 Introduction

1.1 Context and motivation

One of the central questions in the development of multi-parameter persistence theory is to find a proper generalization of the concept of a persistence barcode, which plays a key part in the one-parameter instance of the theory. Given a *one-parameter persistence module*, i.e. a functor M from some subposet $P \subseteq \mathbb{R}$ to the vector spaces over a fixed field \mathbf{k} , the (*persistence*) *barcode* $\text{Dgm } M$ is a multi-set of intervals in P that fully characterizes the module M . Its role in applications is motivated by the fact that $\text{Dgm } M$ provides a compact encoding of the so-called *rank invariant* $\text{Rk } M$, a complete invariant that captures the ranks of the internal morphisms of M , more precisely:

$$\text{Rk } M(s, t) = \text{rank} [M(s) \rightarrow M(t)] \quad \text{for every } s \leq t \in P. \quad (1.1)$$

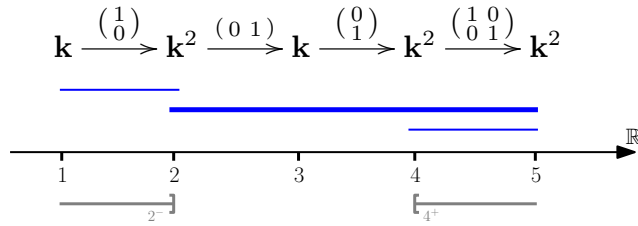
The encoding decomposes $\text{Rk } M$ as a \mathbb{Z} -linear combination of rank invariants of *interval modules*, i.e. indicator modules supported on intervals:

$$\text{Rk } M = \sum_{I \in \text{Dgm } M} \text{Rk } \mathbf{k}_I = \text{Rk} \left(\bigoplus_{I \in \text{Dgm } M} \mathbf{k}_I \right), \quad (1.2)$$

where each interval $I \in \text{Dgm } M$ is considered with multiplicity, and where \mathbf{k}_I denotes the interval module supported on I . Coefficients in the \mathbb{Z} -linear combination are all positive.



19:2 Signed Barcodes for Multi-Parameter Persistence



■ **Figure 1** A one-parameter persistence module M (top) indexed over $\{1, 2, 3, 4, 5\}$, and its barcode (in blue). The corresponding rank decomposition is $\text{Rk } M = \text{Rk } \mathbf{k}_{[1,2]} + \text{Rk } \mathbf{k}_{[2,5]} + \text{Rk } \mathbf{k}_{[4,5]}$. The rank $\text{Rk } M(2, 4) = 1$ is given by the one bar (thickened) connecting the down-set 2^- to the up-set 4^+ .

$$\begin{aligned}
 \text{Rk} \left(\begin{array}{ccccc}
 \mathbf{k} & \xrightarrow{\text{id}} & \mathbf{k} & \xrightarrow{\quad} & 0 \\
 \uparrow \text{id} & & \uparrow \begin{bmatrix} 1 & 0 \end{bmatrix} & & \uparrow \\
 \mathbf{k} & \xrightarrow{\begin{bmatrix} 0 \\ 1 \end{bmatrix}} & \mathbf{k}^2 & \xrightarrow{\begin{bmatrix} 1 & 1 \end{bmatrix}} & \mathbf{k} \\
 \uparrow & & \uparrow \begin{bmatrix} 0 \\ 1 \end{bmatrix} & & \uparrow \text{id} \\
 0 & \xrightarrow{\quad} & \mathbf{k} & \xrightarrow{\quad} & \mathbf{k}
 \end{array} \right) = \text{Rk} \left(\begin{array}{cccc}
 \text{Interval 1} & \oplus & \text{Interval 2} & \oplus & \text{Interval 3} & \oplus & \text{Interval 4} \\
 \text{Interval 5} & & & & & & \text{Interval 6} \\
 \text{Interval 7} & & & & & & \text{Interval 8} \\
 \text{Interval 9} & & & & & & \text{Interval 10}
 \end{array} \right) \\
 - \text{Rk} \left(\begin{array}{cc}
 \text{Interval 11} & \oplus & \text{Interval 12} \\
 \text{Interval 13} & & \text{Interval 14}
 \end{array} \right)
 \end{aligned}$$

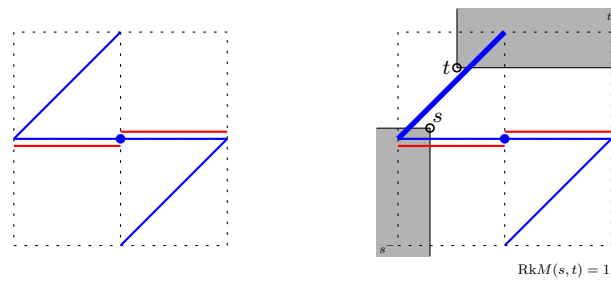
■ **Figure 2** The indecomposable module M on the left does not have the same rank invariant as any direct sum of interval modules on the 3×3 grid. However, $\text{Rk } M$ is equal to the difference between the rank invariants of the two direct sums of interval modules shown on the right. Blue is for intervals counted positively in the decomposition, while red is for intervals counted negatively.

The encoding in (1.2) is unique, i.e. there is no other way to decompose $\text{Rk } M$ as a \mathbb{Z} -linear combination, with positive coefficients, of rank invariants of interval modules. This is because M itself decomposes essentially uniquely as a direct sum of interval modules [6]:

$$M \simeq \bigoplus_{I \in \text{Dgm } M} \mathbf{k}_I. \tag{1.3}$$

Since the intervals in $\text{Dgm } M$ are line segments – possibly closed, open, or half-open, $\text{Dgm } M$ can be represented graphically as an actual barcode (see Figure 1) that reveals the global structure of the rank invariant $\text{Rk } M$, as well as of the module M itself.

Major difficulties arise when trying to generalize the concept of barcode to *multi-parameter persistence modules* – i.e. functors M from \mathbb{R}^d (equipped with the product order) to the vector spaces over \mathbf{k} . Foremost, while a direct-sum decomposition of M into indecomposables still exists and is essentially unique [3], the summands may no longer be interval modules as in (1.3), where intervals in \mathbb{R}^d are defined to be connected convex subsets in the product order. For instance, the module on the left-hand side of Figure 2 is indecomposable yet not an interval module nor even an indicator module – its pointwise dimension is not everywhere ≤ 1 . One may then ask whether rank decompositions such as (1.2) exist nonetheless. The answer is unfortunately negative: still in Figure 2, the module M on the left-hand side does not have the same rank invariant as any direct sum of interval modules, therefore it cannot decompose as in (1.2). Nevertheless, $\text{Rk } M$ can be expressed as the difference between the



■ **Figure 3** Left: the signed barcode corresponding to the rank decomposition of Figure 2. Each bar is the diagonal with positive slope of one of the rectangles involved in the decomposition, with the same color code (blue for positive sign, red for negative sign). Right: computing $\text{Rk} M(s, t)$ for a pair of indices $s \leq t$ – the thick bar is the only one connecting the down-set s^- to the up-set t^+ .

rank invariants of two direct sums of interval modules, as illustrated in the same figure. In other words, $\text{Rk} M$ decomposes as a \mathbb{Z} -linear combination of rank invariants of interval modules, with possibly negative coefficients.

The fact that the signed rank decomposition of the module M in Figure 2 involves only rectangles is not mere chance: the point of our work is to show that such decompositions of the rank invariant (1.1) exist, and furthermore that they are essentially unique – which may not be the case for decompositions of this invariant using larger classes of intervals. Uniqueness in the case of rectangles comes from the known fact that the rank invariant is complete on direct sums of *rectangle modules*, i.e. interval modules supported on rectangles [4, 8].

Rectangles are also interesting because they are entirely determined by their upper bound and lower bound. They therefore allow for an alternative representation of the signed rank decomposition as a *signed barcode*, where each bar is the diagonal (with positive slope) of a particular rectangle in the decomposition, with the same sign. As illustrated in Figure 3, the signed barcode encodes visually the global structure of the rank invariant (1.1), and it gives access to the same information as the signed rank decomposition. For instance, the rank $\text{Rk} M(s, t)$ between a pair of indices $s \leq t$ is given by the number of positive bars that connect the down-set $s^- = \{u \in P \mid u \leq s\}$ to the up-set $t^+ = \{u \in P \mid u \geq t\}$, minus the number of negative bars that connect s^- to t^+ .

1.2 Our setting

We work more generally over a partially ordered set P , considered as a category in the obvious way, and we let \mathbf{k} be an arbitrary but fixed field. Closed rectangles in \mathbb{R}^d now become *closed segments* in P , defined by $\langle s, t \rangle = \{u \in P \mid s \leq u \leq t\}$. *Intervals* in P are defined as non-empty subsets I that are both convex and connected in the partial order. Denote by $\text{Rep} P$ the functor category consisting of all functors $M: P \rightarrow \text{Vec}_{\mathbf{k}}$ where $\text{Vec}_{\mathbf{k}}$ is the category of vector spaces over \mathbf{k} . We shall refer to such a functor M either as a *representation* of P or as a *persistence module* over P , without distinction. Let $\text{rep} P$ be the subcategory of *pointwise finite-dimensional* (pfd) representations, i.e. functors taking their values in the finite-dimensional vector spaces over \mathbf{k} . Denoting by $\{\leq_P\} = \{(a, b) \in P \times P \mid a \leq b\}$ the set of pairs defining the partial order in P , we see the rank invariant (1.1) as a map $\{\leq_P\} \rightarrow \mathbb{N}$ (in the literature, the rank invariant is sometimes defined as a map on $P \times P$ that vanishes outside $\{\leq_P\}$; such a map clearly holds the same information as our rank invariant). For $I \subseteq P$, $M|_I$ denotes $M \circ \iota$ where $\iota: I \hookrightarrow P$ is the canonical inclusion.

Since the usual rank invariant is incomplete, even on the subcategory of *interval-decomposable* modules (i.e. modules isomorphic to direct sums of interval modules), we will consider a generalization of the rank invariant that probes the existence of “features” in the module across arbitrary intervals $I \subseteq P$, not just across closed segments. This generalization is known to be complete on interval-decomposable modules – see [8] or our Proposition 2.8:

► **Definition 1.1.** *Let $M \in \text{Rep } P$. Given an interval $I \subseteq P$, the generalized rank of M over I , denoted by $\text{Rk}_I M$, is defined by:*

$$\text{Rk}_I M = \text{rank} \left[\varprojlim M|_I \rightarrow \varinjlim M|_I \right].$$

Given a collection \mathcal{I} of intervals, the generalized rank invariant of M over \mathcal{I} is the map $\text{Rk}_{\mathcal{I}} M: \mathcal{I} \rightarrow \mathbb{N} \cup \{\infty\}$ defined by $\text{Rk}_{\mathcal{I}} M(I) = \text{Rk}_I M$.

► **Remark 1.2.** To see that Definition 1.1 generalizes the standard rank invariant, observe that when I is a closed segment $\langle i, j \rangle = \{u \in P \mid i \leq u \leq j\}$, we have $\text{Rk}_I M = \text{rank} [M(i) \rightarrow M(j)]$. Hence, taking $\mathcal{I} = \{\langle i, j \rangle \mid i \leq j \in P\} \simeq \{\leq_P\}$ in the above definition gives $\text{Rk}_{\mathcal{I}} M = \text{Rk } M$, the usual rank invariant of M .

In this work we focus on the subcategory $\text{rep}_{\mathcal{I}} P$ of representations M that have a *finite* generalized rank invariant over a fixed collection \mathcal{I} of intervals, i.e. such that $\text{Rk}_I M \in \mathbb{N}$ for all $I \in \mathcal{I}$. Our setting considers in fact arbitrary functions $\mathcal{I} \rightarrow \mathbb{Z}$. Note that $\text{rep}_{\mathcal{I}} P \supseteq \text{rep } P$, since the morphism $\varprojlim M|_I \rightarrow \varinjlim M|_I$ factors through the internal spaces of $M|_I$.

► **Definition 1.3.** *Given a collection \mathcal{I} of intervals in P , and a function $r: \mathcal{I} \rightarrow \mathbb{Z}$, a (signed) rank decomposition of r over \mathcal{I} is given by the following kind of identity:*

$$r = \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}} - \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{S}},$$

where \mathcal{R} and \mathcal{S} are multi-sets of elements taken from \mathcal{I} such that $\mathbf{k}_{\mathcal{R}}$ and $\mathbf{k}_{\mathcal{S}}$ lie in $\text{rep}_{\mathcal{I}} P$, and where by definition $\mathbf{k}_{\mathcal{R}} = \bigoplus_{R \in \mathcal{R}} \mathbf{k}_R$ and $\mathbf{k}_{\mathcal{S}} = \bigoplus_{S \in \mathcal{S}} \mathbf{k}_S$ (note that elements $R \in \mathcal{R}$ and $S \in \mathcal{S}$ are considered with multiplicity). By extension, we call the pair $(\mathcal{R}, \mathcal{S})$ itself a rank decomposition of r over \mathcal{I} . It is minimal if \mathcal{R} and \mathcal{S} are disjoint as multi-sets.

Note that $\text{Rk}_{\mathcal{I}} \mathbf{k}_R(I) = \mathbb{1}_{R \supseteq I}$ for any $I \in \mathcal{I}$ and $R \in \mathcal{R}$ (Proposition 2.1), so $\text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}}(I)$ counts the number of elements in \mathcal{R} that contain I . This number is requested to be finite in the definition ($\mathbf{k}_{\mathcal{R}} \in \text{rep}_{\mathcal{I}} P$): a sufficient condition for this is that \mathcal{R} is *pointwise finite*, i.e. that every index in P belongs to only finitely many elements of \mathcal{R} , for then $\mathbf{k}_{\mathcal{R}} \in \text{rep } P$.

An important consequence of having $\text{Rk}_{\mathcal{I}} \mathbf{k}_R(I) = \mathbb{1}_{R \supseteq I}$ is that adding the same interval $I \in \mathcal{I}$ to both \mathcal{R} and \mathcal{S} does not change the difference $\text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}} - \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{S}}$, so rank decompositions cannot be unique. This motivates the notion of minimal rank decomposition.

1.3 Contributions and structure of the paper

In Section 2 we study the existence and uniqueness of minimal rank decompositions. We show in Theorem 2.9 and Corollary 2.10 that a minimal rank decomposition $(\mathcal{R}, \mathcal{S})$ of a given map $r: \mathcal{I} \rightarrow \mathbb{Z}$ exists as soon as at least one rank decomposition of r exists, that it is always unique, and that it satisfies a universality property justifying its name. To complete the picture, in Corollary 2.5 we provide mild sufficient conditions for the existence of rank decompositions in the first place. Our proofs emphasize the role played by the family of generalized rank invariants $(\text{Rk}_{\mathcal{I}} \mathbf{k}_I)_{I \in \mathcal{I}}$, which acts as a generalized basis (Theorem 2.4).

In Section 3 we reformulate our results in the specific context of multi-parameter persistence. We thus obtain existence and uniqueness results for minimal rank decompositions of finitely presented persistence modules over \mathbb{R}^d (Theorem 3.3), and of pfd persistence modules

over finite grids (Corollary 3.2). In the latter case, we derive an explicit inclusion-exclusion formula to compute the coefficients in the minimal rank decompositions, which generalizes the known formula for counting multiplicities in persistence diagrams in the one-parameter case. We also discuss the stability of the minimal rank decompositions, and propose a metric in which to compare them, based on the matching (pseudo-)distance from [9]. In this metric we show that the minimal rank decompositions are the ones maximizing the distance (Proposition 3.8), and that replacing the modules by their rank decompositions does not expand their pairwise distances (Theorem 3.7).

In Section 4 we introduce the signed barcode as a visual representation of the minimal rank decomposition of the usual rank invariant. We explain how the signed barcode reflects the global structure of the usual rank invariant, and how its role in multi-parameter persistence is similar to the one played by the unsigned barcode in one-parameter persistence. We also discuss its extension to generalized rank invariants, for which it takes the form of a “decorated” signed barcode with similar properties and extra information. The use of these barcodes is illustrated on a practical example coming from 2-parameter clustering.

1.4 Related work

Rank decompositions have strong ties with the concept of *generalized persistence diagram*, introduced by Patel [14] and further studied in [2, 8, 11]. This diagram is defined from the rank invariant via a Möbius inversion, from which our inclusion-exclusion formula for computing the coefficients in the minimal rank decomposition derives. Indeed, in the full version of this paper [5] we show that, whenever it is defined, the generalized persistence diagram does correspond to the minimal rank decomposition. However, our framework allows us to prove the existence and uniqueness of the minimal rank decomposition using direct arguments that: (1.) emphasize the role played by the family of rank invariants of interval modules as a generalized basis for the space of maps $\mathcal{I} \rightarrow \mathbb{Z}$, and (2.) hold in more general settings where the Möbius inversion is not defined. It also allows us to derive stability results for rank decompositions in general (not just minimal ones), in terms of the matching distance d_{match} [9], and to introduce the signed barcodes as a practical graphical representation of minimal rank decompositions – hence of generalized persistence diagrams as well.

2 Rank Decompositions: Existence and Uniqueness

Let P be an arbitrary poset. The following result will be instrumental throughout our analysis. It generalizes [8, Proposition 3.17] by dropping the assumption of local finiteness of the poset P and allowing for generalized ranks, moreover it is given a more direct proof – see the full version of this paper [5]. Note that the result is immediate when working with segments.

► **Proposition 2.1.** *Let \mathcal{R} be a multi-set of intervals of P . Then, for any interval $I \subseteq P$:*

$$\text{Rk}_I(\mathbf{k}_{\mathcal{R}}) = \#\{R \in \mathcal{R} \mid I \subseteq R\}.$$

► **Corollary 2.2.** *Let \mathcal{I} be a collection of intervals in P . For a multi-set \mathcal{R} of intervals, we have that $\mathbf{k}_{\mathcal{R}} \in \text{rep}_{\mathcal{I}} P$ if and only if $\#\{R \in \mathcal{R} \mid I \subseteq R\} < \infty$ for all $I \in \mathcal{I}$.*

2.1 The locally finite case

Let \mathcal{I} be a locally finite collection of intervals in P . That is, for any two comparable intervals in \mathcal{I} , there are only finitely many intervals in \mathcal{I} between the two. We say a map $\mathcal{I} \rightarrow \mathbb{Z}$ has locally finite support if its restriction to the up-set of any element of \mathcal{I} has finite support.

► **Remark 2.3.** For any fixed $I \in \mathcal{I}$, the map $\text{Rk}_{\mathcal{I}} \mathbf{k}_I: J \mapsto \text{Rk}_J \mathbf{k}_I$ has locally finite support, by the description in Proposition 2.1. More generally, for any multi-set \mathcal{R} of elements in \mathcal{I} , if $\mathbf{k}_{\mathcal{R}} \in \text{rep}_{\mathcal{I}} P$ then the map $\text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}}$ has locally finite support: for any fixed $I \in \mathcal{I}$, by Corollary 2.2 \mathcal{R} only contains finitely many elements containing I , and these are the only ones relevant when considering the restriction of $\text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}}$ to the up-set of I .

► **Theorem 2.4.** *Let \mathcal{I} be a locally finite collection of intervals in P . Then any function $r: \mathcal{I} \rightarrow \mathbb{Z}$ with locally finite support can uniquely be written as a (possibly infinite, but pointwise finite) \mathbb{Z} -linear combination of the functions $\text{Rk}_{\mathcal{I}} \mathbf{k}_I$ with $I \in \mathcal{I}$.*

Proof. Existence: Given $I \in \mathcal{I}$ we let $S_I = \{J \supseteq I \mid \exists K \supseteq J \text{ with } r(K) \neq 0\}$. Since r is locally finite, its support restricted to the up-set of I is finite, and so is S_I since \mathcal{I} is locally finite. Now we can define a collection of scalars $\alpha_I \in \mathbb{Z}$ for $I \in \mathcal{I}$, inductively on the size of S_I . If $S_I = \emptyset$ we set $\alpha_I = 0$. Otherwise we set

$$\alpha_I = r(I) - \sum_{J \in S_I \setminus \{I\}} \alpha_J.$$

Note that for $J \in S_I \setminus \{I\}$ we have $S_J \subsetneq S_I$, so the terms in the sum are already defined.

Now, using the description of the map $\text{Rk}_{\mathcal{I}} \mathbf{k}_I$ in Proposition 2.1, one immediately verifies that $r = \sum_{I \in \mathcal{I}} \alpha_I \text{Rk}_{\mathcal{I}} \mathbf{k}_I$. Note in particular that this infinite sum is pointwise finite – on a given interval J the only possibly non-zero terms are the ones in S_J – hence well-defined.

Uniqueness: subtracting two different \mathbb{Z} -linear combinations realizing r from each other, we get a single linear combination $\sum_{I \in \mathcal{I}} \alpha_I \text{Rk}_{\mathcal{I}} \mathbf{k}_I$ with non-zero coefficients which sums up to zero. Note that there is at least one maximal $I \in \mathcal{I}$ such that $\alpha_I \neq 0$, for otherwise the sum would not be defined. It follows, again using Proposition 2.1, that $(\sum_{J \in \mathcal{I}} \alpha_J \text{Rk}_{\mathcal{I}} \mathbf{k}_J)(I) = \alpha_I \neq 0$, contradicting our assumption. ◀

► **Corollary 2.5.** *Let \mathcal{I} be a locally finite collection of intervals in P . Then, for any map $r: \mathcal{I} \rightarrow \mathbb{Z}$ with locally finite support, there is a unique pair \mathcal{R}, \mathcal{S} of disjoint multi-sets of elements of \mathcal{I} such that $r = \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}} - \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{S}}$ and $\mathbf{k}_{\mathcal{R}}, \mathbf{k}_{\mathcal{S}}$ both lie in $\text{rep}_{\mathcal{I}} P$.*

Proof. By Theorem 2.4, there is a unique (possibly infinite, but pointwise finite) \mathbb{Z} -linear combination of functions $r = \sum_{I \in \mathcal{I}} \alpha_I \text{Rk}_{\mathcal{I}} \mathbf{k}_I$. Let then $\mathcal{R} = \{I \in \mathcal{I} \mid \alpha_I > 0\}$ with multiplicities $I \mapsto \alpha_I$, and $\mathcal{S} = \{I \in \mathcal{I} \mid \alpha_I < 0\}$ with multiplicities $I \mapsto |\alpha_I|$. It follows from the pointwise-finiteness of the linear combination that \mathcal{R} and \mathcal{S} satisfy the condition in Corollary 2.2, so in particular $\mathbf{k}_{\mathcal{R}}$ and $\mathbf{k}_{\mathcal{S}}$ lie in $\text{rep}_{\mathcal{I}} P$. ◀

Specializing Theorem 2.4 and Corollary 2.5 to the case where P is finite and $\mathcal{I} = \{\langle i, j \rangle \mid i \leq j \in P\} \simeq \{\leq_P\}$ yields the following results – where $\text{Rk}_{\mathcal{I}}$ becomes the usual rank invariant Rk according to Remark 1.2:

► **Corollary 2.6.** *If P is finite, then the maps $\text{Rk} \mathbf{k}_{\langle a, b \rangle}$ for all $a \leq b \in P$ is a basis of $\mathbb{Z}^{\{\leq_P\}}$.*

► **Corollary 2.7.** *Given a finite poset P , for any map $r: \{\leq_P\} \rightarrow \mathbb{Z}$ there is a unique pair \mathcal{R}, \mathcal{S} of disjoint finite multi-sets of closed segments such that $r = \text{Rk} \mathbf{k}_{\mathcal{R}} - \text{Rk} \mathbf{k}_{\mathcal{S}}$.*

2.2 The general case

We now drop our previous finiteness assumptions and consider arbitrary maps $r: \mathcal{I} \rightarrow \mathbb{Z}$ over an arbitrary collection \mathcal{I} of intervals in an arbitrary poset P . Our first result shows that $\text{Rk}_{\mathcal{I}}$ is a complete invariant when restricted to interval-decomposable representations supported on intervals in \mathcal{I} . In fact, we show that the rank invariant is complete on a slightly larger collection of intervals. This generalizes [8, Theorem 3.14].

► **Proposition 2.8.** *Given a collection \mathcal{I} of intervals in P , let $\widehat{\mathcal{I}} \supseteq \mathcal{I}$ be the collection of limit intervals (which by construction are also intervals):*

$$\widehat{\mathcal{I}} := \left\{ \bigcup_{x \in X} I_x \mid X \text{ totally ordered, } I_x \in \mathcal{I} \text{ and } I_x \subseteq I_y \ \forall x \leq y \in X \right\}.$$

If \mathcal{R} and \mathcal{R}' are two multi-sets of elements in $\widehat{\mathcal{I}}$, such that $\text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}} = \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}'}$ and this common rank invariant is finite, then $\mathcal{R} = \mathcal{R}'$.

Proof. Since the rank of a direct sum is the sum of the ranks we may remove the common elements from \mathcal{R} and \mathcal{R}' , and thus assume that the two multi-sets are disjoint. It follows from the description of multi-sets giving rise to finite ranks in Corollary 2.2 that $\mathcal{R} \cup \mathcal{R}'$ contains at least one maximal element, say J . Without loss of generality we assume $J \in \mathcal{R}$. By definition of $\widehat{\mathcal{I}}$ we have $J = \bigcup_{x \in X} J_x$ with $J_x \in \mathcal{I}$ and $J_x \subseteq J_y$ for all $x \leq y \in X$. Now, by assumption, for every $x \in X$ we have

$$\text{Rk}_{J_x} \mathbf{k}_{\mathcal{R}'} = \text{Rk}_{J_x} \mathbf{k}_{\mathcal{R}} \geq \text{Rk}_{J_x} \mathbf{k}_J,$$

which is at least 1 by Proposition 2.1. It also follows from Proposition 2.1 that, for each $x \in X$, there is some interval $I_x \in \mathcal{R}'$ such that $J_x \subseteq I_x$. Since $\text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}'}$ is finite, Corollary 2.2 says that there are actually only finitely many choices for I_x . Hence, there is an $I \in \mathcal{R}'$ independent of x such that $J_x \subseteq I$ for all $x \in X$. Thus $J \subseteq I$. If this is a proper inclusion then it contradicts the maximality of J , otherwise it contradicts the disjointness of \mathcal{R} and \mathcal{R}' . ◀

We can now show minimal rank decompositions satisfy a universality property when they exist.

► **Theorem 2.9.** *Let $\mathcal{R}, \mathcal{S}, \mathcal{R}^*, \mathcal{S}^*$ be multi-sets of elements of $\widehat{\mathcal{I}}$, whose corresponding representations lie in $\text{rep}_{\mathcal{I}} P$, and such that $\mathcal{R}^* \cap \mathcal{S}^* = \emptyset$. If*

$$\text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}} - \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{S}} = \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}^*} - \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{S}^*}$$

then $\mathcal{R} \supseteq \mathcal{R}^$, $\mathcal{S} \supseteq \mathcal{S}^*$, and $\mathcal{R} \setminus \mathcal{R}^* = \mathcal{S} \setminus \mathcal{S}^*$.*

Proof. Rewriting the equation yields $\text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}} + \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{S}^*} = \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{R}^*} + \text{Rk}_{\mathcal{I}} \mathbf{k}_{\mathcal{S}}$, and by additivity of the rank invariant, $\text{Rk}_{\mathcal{I}}(\mathbf{k}_{\mathcal{R}} \oplus \mathbf{k}_{\mathcal{S}^*}) = \text{Rk}_{\mathcal{I}}(\mathbf{k}_{\mathcal{R}^*} \oplus \mathbf{k}_{\mathcal{S}})$. It follows then by Proposition 2.8 that $\mathcal{R} \cup \mathcal{S}^* = \mathcal{R}^* \cup \mathcal{S}$. As $\mathcal{R}^* \cap \mathcal{S}^* = \emptyset$, we conclude that $\mathcal{R} \supseteq \mathcal{R}^*$, $\mathcal{S} \supseteq \mathcal{S}^*$, and $\mathcal{R} \setminus \mathcal{R}^* = \mathcal{S} \setminus \mathcal{S}^*$. ◀

As an immediate consequence of Theorem 2.9, we obtain uniqueness and conditional existence of minimal rank decompositions:

► **Corollary 2.10.** *The minimal rank decomposition $(\mathcal{R}^*, \mathcal{S}^*)$ of any map $r : \mathcal{I} \rightarrow \mathbb{Z}$ is unique if it exists. Furthermore, it exists as soon as any rank decomposition $(\mathcal{R}, \mathcal{S})$ of r does, being obtained from it by removing common intervals, that is: $(\mathcal{R}^*, \mathcal{S}^*) = (\mathcal{R} \setminus \mathcal{R} \cap \mathcal{S}, \mathcal{S} \setminus \mathcal{R} \cap \mathcal{S})$.*

We also get a connection between the various rank decompositions of a map $\mathcal{I} \rightarrow \mathbb{Z}$:

► **Corollary 2.11.** $\mathcal{R} \cup \mathcal{S}' = \mathcal{R}' \cup \mathcal{S}$ for any rank decompositions $(\mathcal{R}, \mathcal{S}), (\mathcal{R}', \mathcal{S}')$ of $r : \mathcal{I} \rightarrow \mathbb{Z}$.

Proof. Let $(\mathcal{R}^*, \mathcal{S}^*)$ be the minimal rank decomposition of r . By Theorem 2.9, we have $\mathcal{R} = \mathcal{R}^* \cup \mathcal{T}$ and $\mathcal{S} = \mathcal{S}^* \cup \mathcal{T}$ for some finite multi-set \mathcal{T} of elements of \mathcal{I} , while $\mathcal{R}' = \mathcal{R}^* \cup \mathcal{T}'$ and $\mathcal{S}' = \mathcal{S}^* \cup \mathcal{T}'$ for some multi-set \mathcal{T}' . Then, $\mathcal{R} \cup \mathcal{S}' = \mathcal{R}^* \cup \mathcal{S}^* \cup \mathcal{T} \cup \mathcal{T}' = \mathcal{R}' \cup \mathcal{S}$. ◀

19:8 Signed Barcodes for Multi-Parameter Persistence

$$\begin{aligned} \text{Rk}_{\mathcal{I}} \left(\begin{array}{c} \begin{array}{ccccc} k & \xrightarrow{\text{id}} & k & \xrightarrow{\quad} & 0 \\ \uparrow & & \uparrow & & \uparrow \\ \text{id} & & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & & \text{id} \\ k & \xrightarrow{\begin{bmatrix} 1 \\ 0 \end{bmatrix}} & k^2 & \xrightarrow{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}} & k \\ \uparrow & & \uparrow & & \uparrow \\ 0 & \xrightarrow{\quad} & k & \xrightarrow{\text{id}} & k \end{array} \end{array} \right) = \text{Rk}_{\mathcal{I}} \left(\begin{array}{c} \begin{array}{cccc} \text{blue} & & & \\ \text{blue} & & & \\ \text{blue} & & & \\ \text{blue} & & & \end{array} \oplus \begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array} \oplus \begin{array}{cccc} \text{blue} & & & \\ \text{blue} & & & \\ \text{blue} & & & \\ \text{blue} & & & \end{array} \oplus \begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array} \right) \\ - \text{Rk}_{\mathcal{I}} \left(\begin{array}{c} \begin{array}{cccc} \text{red} & & & \\ \text{red} & & & \\ \text{red} & & & \\ \text{red} & & & \end{array} \oplus \begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array} \right) \end{aligned}$$

■ **Figure 4** Minimal rank decomposition of the generalized rank invariant of the module M from Figure 2 over the full collection \mathcal{I} of intervals in the 3×3 grid. Blue is for intervals in \mathcal{R} , red is for intervals in \mathcal{S} .

$$\text{Rk}_{\mathcal{I}} \left(\begin{array}{c} \begin{array}{cccc} & & d & \\ & & f & e \\ & & k & \\ a & c & & \\ & b & & \end{array} \end{array} \right) = \text{Rk}_{\mathcal{I}} \left(\begin{array}{c} \begin{array}{cccc} & & d & \\ & & f & e \\ & & k & \\ a & & & \\ & b & & \end{array} \oplus \begin{array}{cccc} & & d & \\ & & f & e \\ & & k & \\ & & & \\ & b & & \end{array} \right) - \text{Rk}_{\mathcal{I}} \left(\begin{array}{c} \begin{array}{cccc} & & d & \\ & & f & e \\ & & k & \\ & c & & \end{array} \end{array} \right)$$

■ **Figure 5** Taking \mathcal{I} to be the collection of all intervals with one generator and at most two cogenerators (which includes in particular all rectangles), the generalized rank invariant of the interval 2-parameter persistence module $\mathbf{k}_{\mathcal{I}}$ on the left-hand side decomposes minimally as the difference between the generalized rank invariants of the two modules on the right-hand side. Blue is for intervals in \mathcal{R} , red is for intervals in \mathcal{S} .

3 Application to multi-parameter persistence

Here the poset P under consideration is either \mathbb{R}^d , viewed as a product of d copies of the totally ordered real line, or a subset of \mathbb{R}^d – usually \mathbb{Z}^d or some finite grid $\prod_{i=1}^d \llbracket 1, n_i \rrbracket$. The role of segments is played by *rectangles*, i.e. products of 1-d intervals.

3.1 The finite grid case

In this case, Corollary 2.5 reformulates as follows:

► **Corollary 3.1.** *Given an arbitrary collection \mathcal{I} of intervals in a finite grid $G = \prod_{i=1}^d \llbracket 1, n_i \rrbracket \subset \mathbb{R}^d$, the generalized rank invariant $\text{Rk}_{\mathcal{I}} M$ of any pfd persistence module M indexed over G admits a unique minimal rank decomposition $(\mathcal{R}, \mathcal{S})$ over \mathcal{I} .*

Taking \mathcal{I} to be the collection of all closed rectangles in the grid G yields the following reformulation of Corollary 2.7:

► **Corollary 3.2.** *The usual rank invariant of any pfd persistence module M indexed over a finite grid $G = \prod_{i=1}^d \llbracket 1, n_i \rrbracket \subset \mathbb{R}^d$ admits a unique minimal rank decomposition $(\mathcal{R}, \mathcal{S})$, where \mathcal{R} and \mathcal{S} are finite multi-sets of (closed) rectangles in G .*

Figures 4 and 5 illustrate Corollary 3.1, while Figures 2 and 6 illustrate Corollary 3.2.

$$\begin{aligned}
 \text{Rk} \left(\begin{array}{c} \text{---} d \\ \text{---} f \\ \text{---} e \\ \text{---} a \\ \text{---} b \\ \text{---} c \end{array} \right) &= \text{Rk} \left(\begin{array}{c} \text{---} d \\ \text{---} a \\ \text{---} b \\ \text{---} c \\ \text{---} e \\ \text{---} f \end{array} \right) \\
 &= \text{Rk} \left(\begin{array}{c} \text{---} d \\ \text{---} a \\ \text{---} b \\ \text{---} c \\ \text{---} e \\ \text{---} f \end{array} \right) \\
 &\quad - \text{Rk} \left(\begin{array}{c} \text{---} f \\ \text{---} a \\ \text{---} b \\ \text{---} c \\ \text{---} d \\ \text{---} e \end{array} \right)
 \end{aligned}$$

■ **Figure 6** The usual rank invariant of the interval module \mathbf{k}_I on the left-hand side decomposes minimally as the difference between the usual rank invariants of the two rectangle-decomposable modules on the right-hand side. Blue is for rectangles in \mathcal{R} , red is for rectangles in \mathcal{S} .

Computation

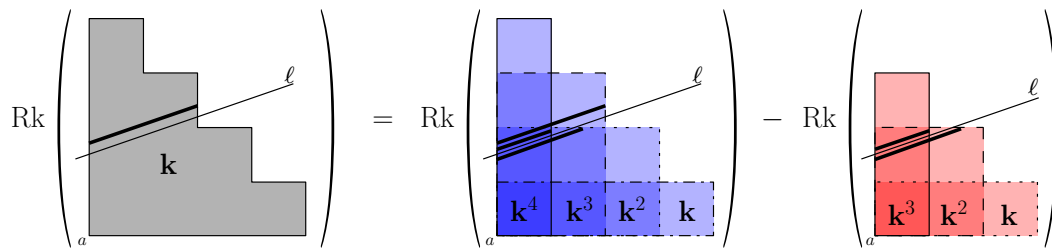
Given $\text{Rk}_{\mathcal{I}} M$, computing its minimal rank decomposition can be done by applying the inclusion-exclusion formula defining the so-called *generalized persistence diagram* of M – see e.g. Definition 3.13 in [8]. Indeed, whenever it exists, the generalized persistence diagram of M coincides with the minimal rank decomposition of $\text{Rk}_{\mathcal{I}}(M)$. This connection happens as both objects derive from the Möbius inversion of the generalized rank invariant – see Section 3 in the full version of our paper [5] for the details. While simple, this approach does not scale up well because the inclusion-exclusion formula must be applied for every interval in the collection \mathcal{I} , whose size can be up to exponential in the size of the indexing grid G , even in two dimensions [1].

In the special case of the usual rank invariant however, the approach scales up reasonably as the number of rectangles is at most quadratic in the size of the grid G . To be more specific, the inclusion-exclusion formula writes as follows in this case, where $\alpha_{\langle s,t \rangle}$ is the coefficient assigned to $\text{Rk} \mathbf{k}_{\langle s,t \rangle}$ in the minimal rank decomposition of $\text{Rk} M$:

$$\forall s \leq t \in G, \alpha_{\langle s,t \rangle} = \sum_{\substack{s' \leq s \\ \|s' - s\|_{\infty} \leq 1}} \sum_{\substack{t' \geq t \\ \|t' - t\|_{\infty} \leq 1}} (-1)^{\|s' - s\|_1 + \|t' - t\|_1} \text{Rk} M(s', t'). \tag{3.1}$$

By applying (3.1) successively to every pair of comparable indices $s \leq t$ in the grid $G = \prod_{i=1}^d \llbracket 1, n_i \rrbracket \subset \mathbb{R}^d$, one computes the minimal rank decomposition of $\text{Rk} M$ in time $O(2^{2d} \#\{\leq_G\})$, assuming constant-time access to the ranks $\text{Rk} M(s', t')$ and constant-time arithmetic operations¹. This bound is in $O(2^{2d} \prod_{i=1}^d n_i^2)$, and when d is fixed, it is linear in the size of the encoding of the usual rank invariant as a map $\{\leq_G\} \rightarrow \mathbb{Z}$. When the module M comes from a simplicial filtration over the grid G with $n = \max_i n_i$ simplices in total, the usual rank invariant itself can be pre-computed and stored, e.g. by naively computing the ranks $\text{Rk} M(s, t)$ for each pair $s \leq t \in G$ independently, which takes $O(n^{2d+\omega})$ time in total, where $2 \leq \omega < 2.373$ is the exponent for matrix multiplication [12]. Adding in the

¹ We are considering an implementation that iterates over the indices s', t' such that $\|s' - s\|_{\infty} \leq 1$ and $\|t' - t\|_{\infty} \leq 1$ by increasing order of the 1-norms $\|s' - s\|_1$ and $\|t' - t\|_1$, so that the 1-norms do not have to be re-computed from scratch at each step. Such an implementation boils down to iterating over the vertices of the unit hypercube in \mathbb{R}^d by increasing order of the number of 1's in their coordinates.



■ **Figure 8** Restricting an interval module \mathbf{k}_I to a monotone line ℓ (left) yields a restriction of the minimal rank decomposition of $\text{Rk } \mathbf{k}_I$ to ℓ (right) – for clarity, the rectangles’ boundaries are shown with different line styles. Here, the restricted rank decomposition is not minimal, as the two interval summands of $\mathbf{k}_{S|_\ell}$ cancel out with two of the three interval summands of $\mathbf{k}_{\mathcal{R}|_\ell}$.

► **Proposition 3.4.** *Let M be a pfd persistence module over \mathbb{R}^d such that the usual rank invariant $\text{Rk } M$ admits a rank decomposition $(\mathcal{R}, \mathcal{S})$. Then, for any monotone line ℓ in \mathbb{R}^d , $(\mathcal{R}, \mathcal{S})$ restricts to a rank decomposition $(\mathcal{R}|_\ell, \mathcal{S}|_\ell)$ of $\text{Rk } M|_\ell$.*

Proof. Observe that $\text{Rk } \mathbf{k}_{\mathcal{R}|_\ell} = \text{Rk } \mathbf{k}_{\mathcal{R} \cap \ell}$ (and likewise for \mathcal{S}). Thus, $\text{Rk } M|_\ell = \text{Rk } \mathbf{k}_{\mathcal{R}|_\ell} - \text{Rk } \mathbf{k}_{\mathcal{S}|_\ell} = \text{Rk } \mathbf{k}_{\mathcal{R} \cap \ell} - \text{Rk } \mathbf{k}_{\mathcal{S} \cap \ell}$. ◀

► **Remark 3.5.** For a general discussion on restrictions of rank decompositions to subsets, see Section 5 in the full version of the paper [5].

Note that the restriction of a minimal decomposition may not be minimal, as different rectangles in \mathcal{R} and \mathcal{S} may restrict to the same 1-d interval – see Figure 8 for an illustration. However, by Corollary 2.10, the minimal rank decomposition $(\mathcal{R}^*, \mathcal{S}^*)$ of $\text{Rk } M|_\ell$ is easily obtained by removing all the common elements in $\mathcal{R}|_\ell$ and $\mathcal{S}|_\ell$. Furthermore, as illustrated in Figure 8 and formalized in the following result, $(\mathcal{R}^*, \mathcal{S}^*)$ actually coincides with the persistence barcode of the one-parameter module $M|_\ell$.

► **Corollary 3.6.** *Every pfd persistence module M over \mathbb{R} admits a unique minimal rank decomposition $(\mathcal{R}, \mathcal{S})$, given by $\mathcal{R} = \text{Dgm } M$, the persistence barcode of M , and $\mathcal{S} = \emptyset$.*

Proof. Follows from (1.3) and Corollary 2.10. ◀

3.4 Stability

We conclude this section by saying a few words about the stability of our rank decompositions. Recall from Corollary 2.11 that we have $\mathbf{k}_{\mathcal{R}} \oplus \mathbf{k}_{\mathcal{S}'} \simeq \mathbf{k}_{\mathcal{R}'} \oplus \mathbf{k}_{\mathcal{S}}$ for any two rank decompositions $(\mathcal{R}, \mathcal{S})$ and $(\mathcal{R}', \mathcal{S}')$ of the same persistence module M , or of two persistence modules M, M' sharing the same (usual) rank invariant. In effect, this is telling us that two rank decompositions are equivalent whenever their ground modules have the same rank invariant. Using the matching (pseudo-)distance d_{match} from [9], we can derive a metric version of this statement (Theorem 3.7), which bounds the defect of equivalence between two rank decompositions in terms of the fibered distance between the rank invariants of their ground modules. Recall that the matching distance between two pfd persistence modules M, N in \mathbb{R}^d is defined as follows:

$$d_{\text{match}}(M, N) = \sup_{\ell \text{ strictly monotone}} \omega(\ell) d_b(M|_\ell, N|_\ell), \tag{3.2}$$

where d_b denotes the usual bottleneck distance between one-parameter persistence modules, and where the weight of ℓ (parametrized as in Section 3.3) is

$$\omega(\ell) = (\min_i t_i - s_i) / (\max_i t_i - s_i) > 0.$$

19:12 Signed Barcodes for Multi-Parameter Persistence

► **Theorem 3.7.** *Let M, M' be pfd persistence modules indexed over \mathbb{R}^d . Then, for any rank decompositions $(\mathcal{R}, \mathcal{S})$ and $(\mathcal{R}', \mathcal{S}')$ of M and M' respectively, we have:*

$$d_{\text{match}}(\mathbf{k}_{\mathcal{R}} \oplus \mathbf{k}_{\mathcal{S}'}, \mathbf{k}_{\mathcal{R}'} \oplus \mathbf{k}_{\mathcal{S}}) \leq d_{\text{match}}(M, M').$$

Proof. Take any strictly monotone line ℓ in \mathbb{R}^d . By (3.2), we have:

$$d_b(M_{|\ell}, M'_{|\ell}) \leq \omega(\ell)^{-1} d_{\text{match}}(M, M').$$

Meanwhile, by Corollary 3.4, $(\mathcal{R}_{|\ell}, \mathcal{S}_{|\ell})$ is a rank decomposition of $M_{|\ell}$, and $(\mathcal{R}'_{|\ell}, \mathcal{S}'_{|\ell})$ is a rank decomposition of $M'_{|\ell}$. By Proposition 2.8, we then have $M_{|\ell} \oplus \mathbf{k}_{\mathcal{S}_{|\ell}} \simeq \mathbf{k}_{\mathcal{R}_{|\ell}}$ and $M'_{|\ell} \oplus \mathbf{k}_{\mathcal{S}'_{|\ell}} \simeq \mathbf{k}_{\mathcal{R}'_{|\ell}}$, from which we deduce:

$$d_b(M_{|\ell}, M'_{|\ell}) \geq d_b(M_{|\ell} \oplus \mathbf{k}_{\mathcal{S}_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}'_{|\ell}}, M'_{|\ell} \oplus \mathbf{k}_{\mathcal{S}_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}'_{|\ell}}) = d_b(\mathbf{k}_{\mathcal{R}_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}'_{|\ell}}, \mathbf{k}_{\mathcal{R}'_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}_{|\ell}}).$$

Combined with the previous equation, this gives:

$$d_b(\mathbf{k}_{\mathcal{R}_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}'_{|\ell}}, \mathbf{k}_{\mathcal{R}'_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}_{|\ell}}) \leq \omega(\ell)^{-1} d_{\text{match}}(M, M').$$

The result follows then by taking the supremum on the left-hand side over all possible choices of strictly monotone lines ℓ . ◀

Note that different choices of rank decompositions $(\mathcal{R}, \mathcal{S})$ and $(\mathcal{R}', \mathcal{S}')$ for M and M' may yield different values for the matching distance $d_{\text{match}}(\mathbf{k}_{\mathcal{R}} \oplus \mathbf{k}_{\mathcal{S}'}, \mathbf{k}_{\mathcal{R}'} \oplus \mathbf{k}_{\mathcal{S}})$. It turns out that the rank decompositions maximizing this distance are precisely the minimal rank decompositions, which therefore also satisfy a universal property in terms of the metric:

► **Proposition 3.8.** *Let M, M' be pfd persistence modules indexed over \mathbb{R}^d . Then, for any rank decompositions $(\mathcal{R}, \mathcal{S})$ and $(\mathcal{R}', \mathcal{S}')$ of M and M' respectively, we have:*

$$d_{\text{match}}(\mathbf{k}_{\mathcal{R}} \oplus \mathbf{k}_{\mathcal{S}'}, \mathbf{k}_{\mathcal{R}'} \oplus \mathbf{k}_{\mathcal{S}}) \leq d_{\text{match}}(\mathbf{k}_{\mathcal{R}^*} \oplus \mathbf{k}_{\mathcal{S}'^*}, \mathbf{k}_{\mathcal{R}'^*} \oplus \mathbf{k}_{\mathcal{S}^*}),$$

where $(\mathcal{R}^*, \mathcal{S}^*)$ and $(\mathcal{R}'^*, \mathcal{S}'^*)$ are the minimal rank decompositions of M and M' respectively – which exist as soon as $(\mathcal{R}, \mathcal{S})$ and $(\mathcal{R}', \mathcal{S}')$ do, by Corollary 2.10.

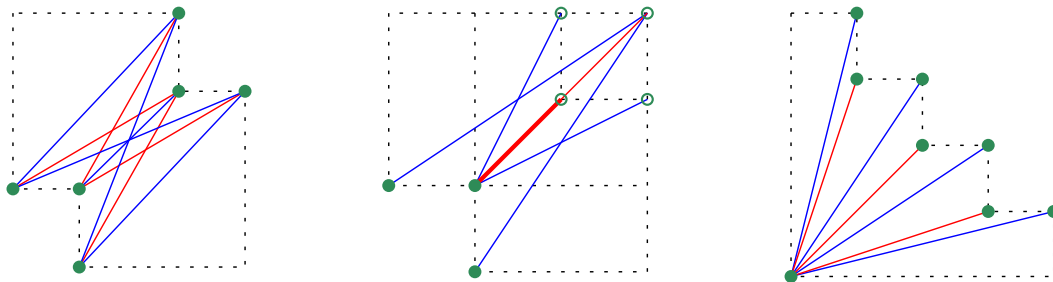
Proof. Let $\mathcal{T} := \mathcal{R} \setminus \mathcal{R}^* = \mathcal{S} \setminus \mathcal{S}^*$, and $\mathcal{T}' := \mathcal{R}' \setminus \mathcal{R}'^* = \mathcal{S}' \setminus \mathcal{S}'^*$. Note that $\mathcal{T}, \mathcal{T}'$ are well-defined by Theorem 2.9. Then, for any strictly monotone line ℓ , we have:

$$\begin{aligned} d_b(\mathbf{k}_{\mathcal{R}_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}'_{|\ell}}, \mathbf{k}_{\mathcal{R}'_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}_{|\ell}}) &= d_b(\mathbf{k}_{\mathcal{R}^*_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}'^*_{|\ell}} \oplus \mathbf{k}_{\mathcal{T}_{|\ell}} \oplus \mathbf{k}_{\mathcal{T}'_{|\ell}}, \mathbf{k}_{\mathcal{R}'^*_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}^*_{|\ell}} \oplus \mathbf{k}_{\mathcal{T}_{|\ell}} \oplus \mathbf{k}_{\mathcal{T}'_{|\ell}}) \\ &\leq d_b(\mathbf{k}_{\mathcal{R}^*_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}'^*_{|\ell}}, \mathbf{k}_{\mathcal{R}'^*_{|\ell}} \oplus \mathbf{k}_{\mathcal{S}^*_{|\ell}}). \end{aligned}$$

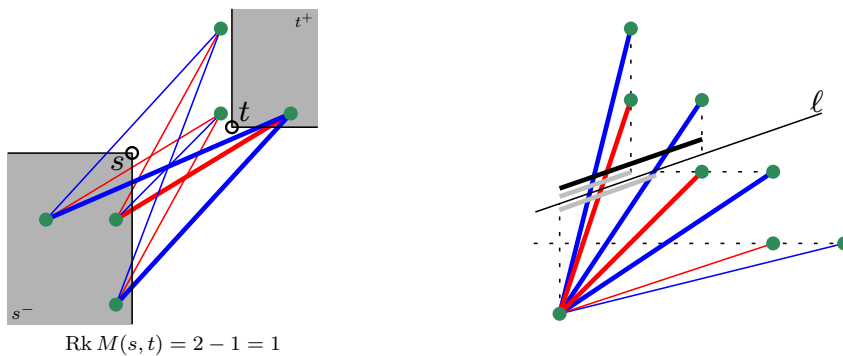
The result follows then after multiplying by $\omega(\ell)$ and taking the supremum on both sides over all possible choices of strictly monotone lines ℓ . ◀

4 Signed barcodes and prominence diagrams

In the context of topological data analysis, the minimal rank decomposition $(\mathcal{R}, \mathcal{S})$ of $\text{Rk } M$ encodes visually the structure of the rank invariant of $M: \mathbb{R}^d \rightarrow \text{Vec}_{\mathbf{k}}$. However, representing rectangles as rectangles quickly leads to arrangements that are hard to read – see e.g. Figure 8.



■ **Figure 9** From left to right: signed barcodes corresponding to the usual rank decompositions of Figures 6, 7, and 8 respectively. Blue bars are diagonals of rectangles in \mathcal{R} and therefore counted positively, while red bars are diagonals of rectangles in \mathcal{S} and therefore counted negatively. The bars' endpoints are marked in green (as a solid dot when the endpoint lies in the rectangle, as a circled dot when it does not – e.g. when it lies at infinity), to discriminate them from intersections. The thick red line segment in the center picture shows the overlap between a shorter red bar and a longer red bar sharing the same lower endpoint and slope.

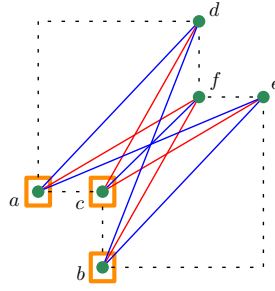


■ **Figure 10** Left: computing $\text{Rk } M(s, t)$ for a pair of indices $s \leq t$ – the thick bars are the ones connecting the down-set s^- to the up-set t^+ . Right: restricting the minimal rank decomposition of $\text{Rk } M$ to a strictly monotone line ℓ – the thick blue and red bars are the ones projecting to non-empty bars along ℓ , and among those projections, the thick gray bars get cancelled out during the simplification while the thick black bar remains in the barcode of $M|_\ell$.

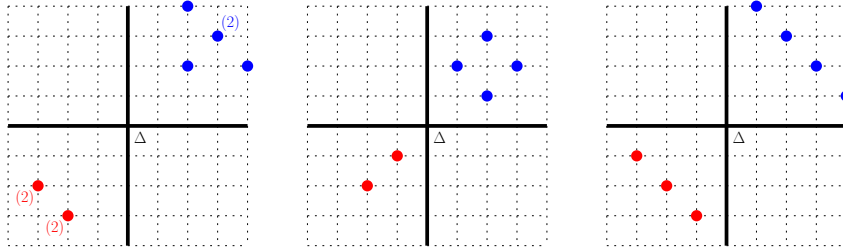
4.1 Signed barcodes

An alternate representation of the rectangles is by their diagonal with positive slope. We call this representation the *signed barcode* of $\text{Rk } M$, where each bar is the diagonal (with positive slope) of a particular rectangle in \mathcal{R} or \mathcal{S} , and where the sign is positive for bars coming from \mathcal{R} and negative for bars coming from \mathcal{S} – see Figure 9 for an illustration. Like the rectangles, the bars are considered with multiplicity. The signed barcode of $\text{Rk } M$ gives direct access to the same pieces of information as the rectangular representation, as shown in Figure 10. Furthermore, the signed barcode makes it possible to visually grasp the global structure of the usual rank invariant $\text{Rk } M$, and in particular, to infer the directions along which topological features have the best chances to persist.

When the collection \mathcal{I} of intervals under consideration contains more than just the rectangles, the intervals involved in the corresponding minimal rank decomposition $(\mathcal{R}, \mathcal{S})$ of M are no longer described by a single diagonal. Nevertheless, each interval $I \in \mathcal{R} \sqcup \mathcal{S}$ is still uniquely described by the signed barcode of the corresponding interval module \mathbf{k}_I . We



■ **Figure 11** Decorated signed barcode corresponding to the generalized rank decomposition $(\mathcal{R}, \mathcal{S})$ of Figure 5. The orange squares indicate how the bars are grouped together according to their originating element $I \in \mathcal{R} \sqcup \mathcal{S}$.



■ **Figure 12** The signed prominence diagrams corresponding to the signed barcodes of Figure 9, in the same order. Blue dots correspond to blue bars (hence to rectangles in \mathcal{R}), while red dots correspond to red bars (hence to rectangles in \mathcal{S}). Multiplicities differing from 1 are indicated explicitly. The union Δ of the two coordinate axes plays the role of the diagonal.

can then collate all these signed barcodes together, negating the ones coming from intervals in \mathcal{S} , which yields a global *decorated* signed barcode for $\text{Rk}_{\mathcal{I}} M$, where the decoration groups the bars according to which element $I \in \mathcal{R} \sqcup \mathcal{S}$ they originate from – see Figure 11.

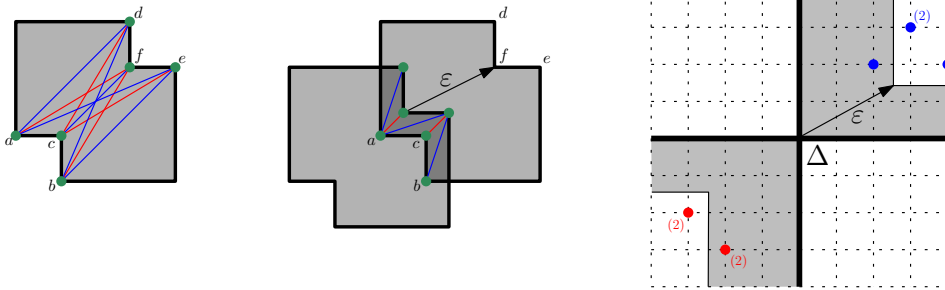
4.2 Signed prominence diagrams

To each bar with endpoints $s \leq t$ in the (undecorated) signed barcode of $\text{Rk} M$, we can associate its *signed prominence*, which is the d -dimensional vector $t - s$ if the bar corresponds to a rectangle in \mathcal{R} , or $s - t$ if the bar corresponds to a rectangle in \mathcal{S} . We call *signed prominence diagram* of M the resulting collection of vectors in \mathbb{R}^d – see Figure 12.

In a signed prominence diagram, the union Δ of the hyperplanes perpendicular to the coordinate axes and passing through the origin plays the role of the diagonal: a bar whose signed prominence lies close to Δ can be viewed as noise, whereas a bar whose signed prominence lies far away from Δ can be considered significant for the structure of $\text{Rk} M$. The right way to formalize this intuition is via smoothings, as in the one-parameter case.

► **Definition 4.1.** Given $\varepsilon \in \mathbb{R}_{\geq 0}^d$, the ε -shift $M[\varepsilon]$ is the persistence module defined pointwise by $M[\varepsilon](t) = M(t + \varepsilon)$ and $M[\varepsilon](s \leq t) = M(s + \varepsilon \leq t + \varepsilon)$. There is a canonical morphism of persistence modules $M \rightarrow M[\varepsilon]$, whose image M^ε is called the ε -smoothing of M .

► **Example 4.2.** The ε -shift of a rectangle module \mathbf{k}_R is $\mathbf{k}_{R-\varepsilon}$, where by definition $R - \varepsilon$ is the shifted rectangle $\{t - \varepsilon \mid t \in R\}$. The ε -smoothing of \mathbf{k}_R is $\mathbf{k}_{R^\varepsilon}$, where by definition R^ε is the rectangle $R \cap (R - \varepsilon)$, obtained from R by shifting its upper-right corner by $-\varepsilon$.



■ **Figure 13** Behavior of the signed barcode and prominence diagram under ε -smoothing. Left: the input module M from Figure 6, overlaid with its signed barcode. Center: the ε -smoothing M^ε of M (in dark gray), overlaid with its own signed barcode – obtained by shifting the right endpoints in the signed barcode of M by $-\varepsilon$. Right: effect of the ε -smoothing on the signed prominence diagram.

As it turns out, rank decompositions of the usual rank invariant commute with smoothings:

► **Lemma 4.3.** *If $(\mathcal{R}, \mathcal{S})$ is a rank decomposition of $\text{Rk } M$, then, for any $\varepsilon \in \mathbb{R}_{\geq 0}^d$, the pair $(\mathcal{R}^\varepsilon, \mathcal{S}^\varepsilon)$ where $\mathcal{R}^\varepsilon = \{R^\varepsilon \mid R \in \mathcal{R}\}$ and $\mathcal{S}^\varepsilon = \{S^\varepsilon \mid S \in \mathcal{S}\}$ is a rank decomposition of $\text{Rk } M^\varepsilon$. If $(\mathcal{R}, \mathcal{S})$ is minimal, then so is $(\mathcal{R}^\varepsilon, \mathcal{S}^\varepsilon)$ after removing the empty rectangles from \mathcal{R}^ε and \mathcal{S}^ε .*

See our full version [5] for an elementary proof of this result, which says that the effect of ε -smoothing M on its signed barcode is to shift the right endpoints of the bars by $-\varepsilon$, removing those bars for which the shifted right endpoint is no longer greater than or equal to the left endpoint. The effect on its signed prominence diagram is to shift the positive vectors by $-\varepsilon$ and the negative vectors by ε , removing those vectors that cross Δ . Alternatively, one can inflate Δ by ε , and remove the vectors that lie in the inflated Δ , as illustrated in Figure 13.

4.3 A practical example: two-parameter clustering

We consider the point set P shown in Figure 14, which consists of $N = 90$ planar points sampled from three different Gaussian distributions. We build the *Vietoris–Rips bifiltration* from this dataset, given by $\text{VR}(P)_{r,s} := \text{VR}(f_\varepsilon^{-1}(-\infty, s])_r$, where $\text{VR}(\cdot)_r$ denotes the usual Vietoris–Rips complex of parameter r , and where $f_\varepsilon: P \rightarrow \mathbb{R}$ is a local co-density estimator:

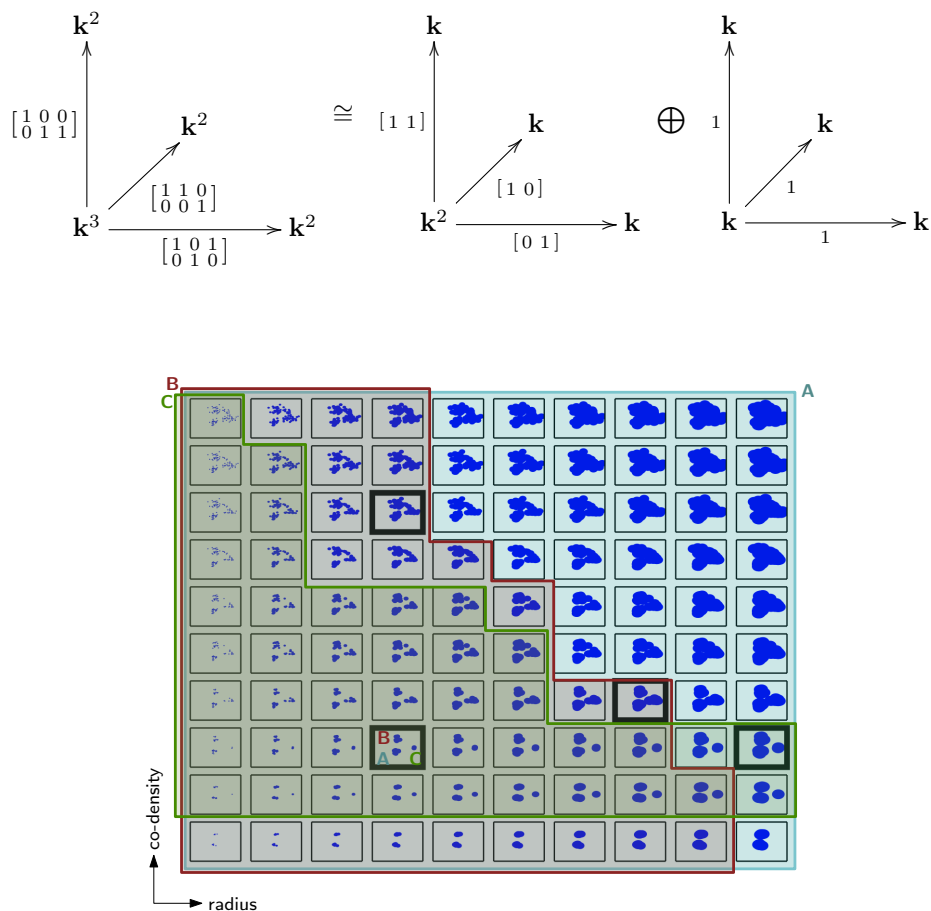
$$f_\varepsilon(p) = \#\{q \in P : d(p, q) > \varepsilon\}, \quad \text{for a fixed parameter } \varepsilon \geq 0.$$

As the Vietoris–Rips complex $\text{VR}(P)_{r,s}$ can be hard to visualize, we replace it in our plots by a proxy union of balls, $U_{r,s} = \left\{ z \in \mathbb{R}^2 : \min_{p \in P, f_\varepsilon(p) \leq s} \|p - z\| \leq r/2 \right\}$, which is known to be interleaved multiplicatively with it.

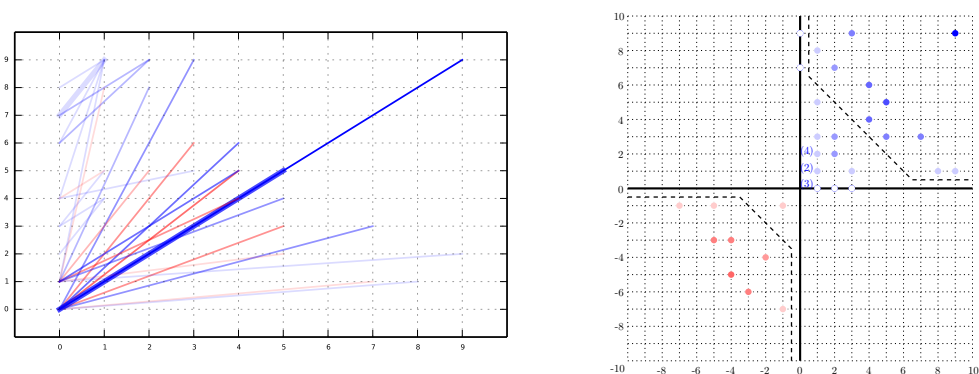
Applying simplicial 0-homology with coefficients in the field \mathbb{Z}_2 yields a bipersistence module $M: M(r, s) = H_0(\text{VR}(P)_{r,s})$. In practice we discretize M over a 10×10 regular grid G , which we identify with the grid $\{0, 1, \dots, 9\} \times \{0, 1, \dots, 9\}$ in our plots. We know that², if $(\mathcal{R}, \mathcal{S})$ is a rank decomposition of M , then $(\mathcal{R}|_G, \mathcal{S}|_G)$ is a rank decomposition of $M|_G$. Note that the persistence module thus obtained is not interval-decomposable. Geometrically, this is due to three clusters A, B, C merging in three different ways at incomparable grades, as shown in the highlighted squares of Figure 14, so that we have the following diagrams:

² This comes from an extension of Proposition 3.4 to lattices, proven in the full version of the paper [5].

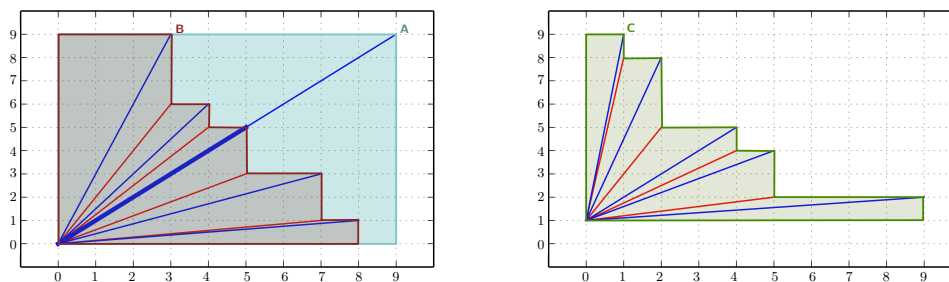
19:16 Signed Barcodes for Multi-Parameter Persistence



■ **Figure 14** The bifiltration in our experiment. The highlighted black squares show that three clusters (named A, B, C) merge in three different ways at incomparable scales. The lifespan of each one of these three clusters is marked by an interval with matching color.



■ **Figure 15** Left: signed barcode of our experiment over the 10×10 grid G , where thicker bars overlap with another bar. Right: corresponding prominence diagram, where the bars coming from the lifespans of A, B, C are separated from the rest of the bars by the dashed curves. Each bar with endpoints $s \leq t$ in the barcode (and diagram) has an intensity proportional to $\min\{t_x - s_x, t_y - s_y\}$.



■ **Figure 16** Lifespans of A, B (left) and C (right) in the signed barcode.

The resulting signed barcode is shown in Figure 15. As expected, the lifespans of the three clusters A, B, C appear as separate subsets of the bars, as shown in Figure 16. Checking whether one of these three subsets does correspond to the lifespan of some feature can be done by computing the coefficient assigned to the corresponding interval in the generalized rank decomposition of M . The decorated barcode would provide this information as well.

References

- 1 Hideto Asashiba, Mickaël Buchet, Emerson G. Escolar, Ken Nakashima, and Michio Yoshiwaki. On interval decomposability of 2d persistence modules, 2018. [arXiv:1812.05261](#).
- 2 Hideto Asashiba, Emerson G. Escolar, Ken Nakashima, and Michio Yoshiwaki. On approximation of 2d persistence modules by interval-decomposables. *arXiv preprint*, 2019. [arXiv:1911.01637](#).
- 3 Magnus Botnan and William Crawley-Boevey. Decomposition of persistence modules. *Proceedings of the American Mathematical Society*, 148(11):4581–4596, 2020.
- 4 Magnus Bakke Botnan, Vadim Lebovici, and Steve Oudot. On Rectangle-Decomposable 2-Parameter Persistence Modules. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. [doi:10.4230/LIPIcs.SoCG.2020.22](#).
- 5 Magnus Bakke Botnan, Steffen Oppermann, and Steve Oudot. Signed barcodes for multi-parameter persistence via rank decompositions and rank-exact resolutions. *arXiv preprint*, 2021. [arXiv:2107.06800](#).
- 6 William Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules. *Journal of Algebra and its Applications*, 14(05):1550066, 2015.
- 7 Tamal K Dey and Cheng Xin. Generalized persistence algorithm for decomposing multi-parameter persistence modules. *arXiv preprint*, 2019. [arXiv:1904.03766](#).
- 8 Woojin Kim and Facundo Memoli. Generalized persistence diagrams for persistence modules over posets. *arXiv preprint*, 2018. [arXiv:1810.11517](#).
- 9 Claudia Landi. The rank invariant stability via interleavings. In *Research in computational topology*, pages 1–10. Springer, 2018.
- 10 Michael Lesnick and Matthew Wright. Interactive visualization of 2-d persistence modules. *arXiv preprint*, 2015. [arXiv:1512.00180](#).
- 11 Alexander McCleary and Amit Patel. Edit distance and persistence diagrams over lattices, 2021. [arXiv:2010.07337](#).

19:18 Signed Barcodes for Multi-Parameter Persistence

- 12 Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 216–225. ACM, 2011.
- 13 Dmitriy Morozov. *Homological illusions of persistence and stability*. PhD thesis, Duke University, 2008.
- 14 Amit Patel. Generalized persistence diagrams. *Journal of Applied and Computational Topology*, 1(3):397–419, 2018.

Dynamic Time Warping Under Translation: Approximation Guided by Space-Filling Curves

Karl Bringmann ✉

Universität des Saarlandes, Saarbrücken, Germany

Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Sándor Kisfaludi-Bak ✉

Aalto University, Espoo, Finland

Marvin Künnemann

Institute for Theoretical Studies, ETH Zürich, Switzerland

Dániel Marx ✉ 

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

André Nusser ✉ 

BARC, University of Copenhagen, Denmark

Abstract

The Dynamic Time Warping (DTW) distance is a popular measure of similarity for a variety of sequence data. For comparing polygonal curves π, σ in \mathbb{R}^d , it provides a robust, outlier-insensitive alternative to the Fréchet distance. However, like the Fréchet distance, the DTW distance is not invariant under translations. Can we efficiently optimize the DTW distance of π and σ under arbitrary translations, to compare the curves' *shape* irrespective of their absolute location?

There are surprisingly few works in this direction, which may be due to its computational intricacy: For the Euclidean norm, this problem contains as a special case the geometric median problem, which provably admits no exact algebraic algorithm (that is, no algorithm using only addition, multiplication, and k -th roots). We thus investigate exact algorithms for non-Euclidean norms as well as approximation algorithms for the Euclidean norm.

For the L_1 norm in \mathbb{R}^d , we provide an $\mathcal{O}(n^{2(d+1)})$ -time algorithm, i.e., an exact polynomial-time algorithm for constant d . Here and below, n bounds the curves' complexities. For the Euclidean norm in \mathbb{R}^2 , we show that a simple problem-specific insight leads to a $(1 + \varepsilon)$ -approximation in time $\mathcal{O}(n^3/\varepsilon^2)$. We then show how to obtain a subcubic $\tilde{\mathcal{O}}(n^{2.5}/\varepsilon^2)$ time algorithm with significant new ideas; this time comes close to the well-known quadratic time barrier for computing DTW for fixed translations. Technically, the algorithm is obtained by speeding up repeated DTW distance estimations using a dynamic data structure for maintaining shortest paths in weighted planar digraphs. Crucially, we show how to traverse a candidate set of translations using space-filling curves in a way that incurs only few updates to the data structure.

We hope that our results will facilitate the use of DTW under translation both in theory and practice, and inspire similar algorithmic approaches for related geometric optimization problems.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Dynamic Time Warping, Sequence Similarity Measures

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.20

Related Version *Full Version*: <https://arxiv.org/abs/2203.07898>

Funding *Karl Bringmann*: This work is part of the project TIPEA that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 850979).

Sándor Kisfaludi-Bak: Part of this research was conducted while the author was at the Max Planck Institute for Informatics, and part of it while he was at the Institute for Theoretical Studies, ETH Zürich.



© Karl Bringmann, Sándor Kisfaludi-Bak, Marvin Künnemann, Dániel Marx, and André Nusser;

licensed under Creative Commons License CC-BY 4.0

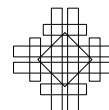
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 20; pp. 20:1–20:17

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Marvin Künnemann: Research supported by Dr. Max Rössler, by the Walter Haefner Foundation, and by the ETH Zürich Foundation.

Dániel Marx: Research supported by the European Research Council (ERC) consolidator grant No. 725978 SYSTEMATICGRAPH.

André Nusser: Part of this research was conducted while the author was at Saarbrücken Graduate School of Computer Science and Max Planck Institute for Informatics. The author is supported by the VILLUM Foundation grant 16582.

Acknowledgements We thank Christian Wulff-Nilsen for making us aware of an improved offline dynamic shortest paths data structure.

1 Introduction

Fast algorithms for computing similarity measures for sequence data enable a number of applications such as signature/handwriting recognition [39, 18], map matching [8, 37], analysis of GPS tracking data [9] and many more. For polygonal curves in \mathbb{R}^d , a popular measure is the Fréchet distance [4, 20] – we refer to [25] for an overview over the extensive literature. Unfortunately, the Fréchet distance is very sensitive to outliers, as the distance value may easily be dominated by erroneous samplings of the curves. Consequently, some contexts would profit from a measure that is more robust to outliers, such as the average/integral Fréchet distance (see [14, 31]) or the well-known dynamic time warping (DTW) distance. The DTW distance is particularly popular for audio sequences (such as speech recognition) and other domains, but has seen an increasing number of uses for geometric curves [33, 39, 18, 38, 9].

Given two polygonal curves $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_m)$ in \mathbb{R}^d , their DTW distance $d_{\text{DTW}}(\pi, \sigma)$ can be defined as follows: We imagine a dog walking on π and its owner walking on σ . Both owner and dog start at the beginning of their curves, and in each step independently decide to either stay in place or jump to the next vertex, until both of them have reached the end of their curves. Formally, this yields a traversal $T = ((i_1, j_1), \dots, (i_t, j_t))$ where $i_1 = j_1 = 1$, $i_t = n$, $j_t = m$ and $(i_{\ell+1}, j_{\ell+1}) \in \{(i_\ell + 1, j_\ell), (i_\ell, j_\ell + 1), (i_\ell + 1, j_\ell + 1)\}$. We define the cost of this traversal as the sum of distances of dog and owner during the traversal, i.e., $\sum_{\ell=1}^t \|\pi_{i_\ell} - \sigma_{j_\ell}\|$. The corresponding DTW distance $d_{\text{DTW}}(\pi, \sigma)$ is defined as the minimum cost of such a traversal.¹ Note that this measure depends on the metric space we use for our curves π, σ . For any metric that we can evaluate in constant time, a simple dynamic programming approach computes $d_{\text{DTW}}(\pi, \sigma)$ in time $O(nm)$, i.e., time $O(n^2)$ when both curves have at most n vertices. While one can achieve mild improvements over this running time [23], one can rule out $O(n^{2-\varepsilon})$ -time algorithms under the Strong Exponential Time Hypothesis, already for curves in \mathbb{R} [1, 11]. Even for constant-factor approximations, no strongly subquadratic algorithms are known, see [30] for (sub)polynomial approximation guarantees and [3, 38] for approximation algorithms on restricted input models.

Unfortunately, the DTW distance is not *translation-invariant*: Distant copies of the same curve may have a much larger distance than differently shaped curves that stay close to each other, see Figure 1. For certain curve similarity applications such as signature recognition, it is thus frequently argued (sometimes implicitly) that a translation-invariant measure is desirable, see e.g. [33, 19, 35, 39, 17].

¹ For comparison, to obtain the discrete Fréchet distance of π and σ , we would minimize, over all traversals T , the *maximum* distance of the dog and its owner during T – one may think of the smallest leash length required to connect dog and owner while traversing their curves.



■ **Figure 1** Curves with similar shape but large DTW distance (left) and different shape but small DTW distance (right).

Arguably the most natural way to make any curve distance measure translation-invariant is to take its minimum under translations of the curves: correspondingly, DTW under translation is defined as $d_{\text{DTW}}^T(\pi, \sigma) := \min_{\tau \in \mathbb{R}^d} d_{\text{DTW}}(\pi, \sigma + \tau)$. Unfortunately, for computing this translation-invariant measure, much less is known than, e.g., for the Fréchet distance under translation. This state of the art, which we review below, is the starting point for our work.

Translation-invariant curve similarity measures. For the continuous Fréchet distance, the earliest algorithmic work studying its translation-invariant version dates back to 2001 [19, 5], with algorithms running in time $\tilde{O}(n^{10})$ and $\tilde{O}(n^8)$, respectively. For the discrete Fréchet distance under translation, algorithms have been improved from $\tilde{O}(n^6)$ [26], via $\tilde{O}(n^5)$ [7], to $\tilde{O}(n^{4.667})$ [13], with a conditional lower bound of $n^{4-o(1)}$ based on the Strong Exponential Time Hypothesis [13]. These theoretical results have been complemented by an algorithm engineering study [12]. Approximation algorithms have been given by [19, 5], including a $(1 + \varepsilon)$ -approximation in time $O(n^2/\varepsilon^2)$. Other works study related settings, such as more general transformations than translations [36, 32], or data structure variants [24].

Unfortunately, we are not aware of algorithmic works with rigorous analyses for DTW under translation, but only heuristic approaches or works on related but different measures. Qiao and Yasuhara [39] experimentally evaluate an iterative method for DTW distance under transformations including translation, rotation and scaling, but provide no theoretical guarantees. Vlachos, Kollios, and Gunopulos [35] study a closely related measure, a variation of the Longest Common Subsequence distance for geometric curves that is translation-invariant. This measure is similar to the DTW distance under translation using a binary distance metric with $d(x, y) = 0$ if $\|x - y\|_\infty \leq \varepsilon$ and $d(x, y) = 1$ otherwise. For their measure, Vlachos et al. provide both exact and approximation algorithms. Munich and Perona [33] define another translation-invariant measure that roughly speaking minimizes differences in direction and velocity changes over traversals of the curves. Efrat, Fan, and Venkatasubramanian [18] study further variants of this measure.

One of the reasons for this lack of rigorous algorithmic work for DTW under translation may be its computational intricacy: Already when $\pi = (\pi_1, \dots, \pi_n)$ is a polygonal curve in \mathbb{R}^d and $\sigma = (\sigma_1)$ consists of a single point $\sigma_1 \in \mathbb{R}^d$, we obtain the geometric median problem as a special case. Specifically, the task simplifies to finding a point $x \in \mathbb{R}^d$ such that $\sum_{i=1}^n \|\pi_i - x\|$ is minimized. This problem provably has no exact algebraic algorithm already for $n = 5$ and $d = 2$ [6] (that is, no algorithm using only addition, multiplication, and k -th roots). We refer to [15] for a recent near-linear time approximation algorithm and an overview of the literature on geometric median. By this lack of an exact, efficient algorithm for geometric median, we can thus hardly expect to solve DTW under translation in Euclidean spaces *exactly*. This motivates to study the problem for norms other than Euclidean, as well as to study approximation algorithms for the Euclidean norm.

1.1 Our results

Exact algorithms for non-Euclidean norms. For the L_1 norm in \mathbb{R}^d , we give a polynomial-time exact algorithm whenever d is constant.

► **Theorem 1.** *For the L_1 -norm in \mathbb{R}^d we can solve DTW under translation in time $\mathcal{O}(n^{2(d+1)})$.*

Since in \mathbb{R}^2 we can transform the L_∞ norm to the L_1 norm by rotating the input by $\frac{\pi}{2}$ and scaling by $1/\sqrt{2}$, this also yields an $\mathcal{O}(n^6)$ time algorithm for L_∞ in \mathbb{R}^2 . We prove the result in the full version.

Approximation algorithms for the Euclidean norm. The main focus in this paper is DTW under translation in the Euclidean plane. Since there is no exact algebraic algorithm due to the special case of geometric median, we focus on developing an *approximation* algorithm.

As a first baseline, we observe that DTW under translation is at least as hard to compute as DTW for a fixed translation, even for approximation (we prove this in the full version). Since exactly computing DTW for a fixed translation requires time $n^{2-o(1)}$ under the Strong Exponential Time Hypothesis [1, 11], and no subquadratic-time constant-factor approximation algorithm is known, the best we could hope for with current techniques would be a $f(1/\varepsilon)n^2$ -time algorithm. Can we reach this baseline or does optimizing over translations in \mathbb{R}^2 increase the problem’s complexity (and if so, by how much)?

For the discrete Fréchet distance, optimizing over a translation increases the time complexity from $n^{2\pm o(1)}$ [20, 10] to at least $n^{4-o(1)}$ and at most $\mathcal{O}(n^{4.667})$ [13] (where the lower bounds are based on the Strong Exponential Time Hypothesis). For $(1 + \varepsilon)$ -approximations, a simple algorithm indeed manages to match the baseline of $\mathcal{O}(n^2/\varepsilon^2)$, see [5]. Does the same hold true for the DTW distance?

Similar arguments to [5] only achieve an $\tilde{\mathcal{O}}(n^4/\varepsilon^2)$ time bound for DTW under translation. Using an insight specific to the nature of the DTW distance, we present a surprisingly simple $\tilde{\mathcal{O}}(n^3/\varepsilon^2)$ time algorithm. We describe both approaches in Section 1.2. Our most important contribution is to obtain a *subcubic* $\tilde{\mathcal{O}}(n^{2.5}/\varepsilon^2)$ time bound via a sophisticated approach that exploits geometric arguments (specifically, a traversal via space-filling curves) to reduce our problem to maintaining shortest paths in a dynamically changing directed grid graph.

► **Theorem 2.** *For the Euclidean norm in \mathbb{R}^2 , we can solve $(1 + \varepsilon)$ -approximate DTW under translation in time $\tilde{\mathcal{O}}(n^{2.5}/\varepsilon^2)$.*

Our techniques strengthen the paradigm of using dynamic algorithms for geometric optimization problems, for which we see a growing number of applications (besides classical examples such as [34], see, e.g., recent work for the Fréchet distance under translation [7, 13] or polygon placement [29]). Finally, only a sublinear factor of $\tilde{\mathcal{O}}(\sqrt{n})$ to the baseline of $\tilde{\mathcal{O}}(n^2/\varepsilon^2)$ remains, which one might hope to decrease by further developing our ideas.

1.2 Technical overview

In this section, we describe the main ideas for our approximation algorithm for DTW under translation. To keep this exposition as simple as possible, we assume that both curves have the same complexity; let these curves be denoted by $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_n)$ throughout this section. The proof in Section 3 gives the slightly more detailed arguments for possibly different complexities of the curves. We start off with a simple algorithm that achieves a rather modest approximation guarantee: Let $\tau_{\text{start}} := \pi_1 - \sigma_1$ denote the

translation of σ that aligns the first points of π and $\sigma + \tau$. It is straightforward to prove that the resulting DTW distance $\delta_{\text{start}} := d_{\text{DTW}}(\pi, \sigma + \tau_{\text{start}})$ yields a $2n$ -approximation to DTW under translation, i.e., $d_{\text{DTW}}^T(\pi, \sigma) \in [\delta_{\text{start}}/(2n), \delta_{\text{start}}]$. This follows from the fact that $d_{\text{DTW}}(\pi, \sigma + \tau)$ is $(2n - 1)$ -Lipschitz with respect to τ , and that $\tau^* := \operatorname{argmin}_{\tau} d_{\text{DTW}}(\pi, \sigma + \tau)$ satisfies $\|\tau_{\text{start}} - \tau^*\| \leq d_{\text{DTW}}^T(\pi, \sigma)$, see Lemma 4. (Analogous arguments are known to give a 2-approximation for the Fréchet distance under translation [5, 12].)

With this rough approximation, the main task for approximating DTW under translation is to design an approximate decider with the following guarantee: Given the polygonal curves π, σ , a threshold $\delta > 0$ and approximation parameter $\varepsilon > 0$, output a verdict “ $d_{\text{DTW}}^T(\pi, \sigma) \leq (1 + \varepsilon)\delta$ ” or “ $d_{\text{DTW}}^T(\pi, \sigma) > \delta$ ” in time $T(n, \varepsilon)$. In any case, the returned verdict has to be correct, i.e., if $\delta < d_{\text{DTW}}^T(\pi, \sigma) \leq (1 + \varepsilon)\delta$ any output is admissible, otherwise it is uniquely determined. Given such an approximate decider, it is straightforward to obtain a $(1 + \varepsilon)$ -approximation algorithm with running time $\mathcal{O}(T(n, \varepsilon/3) \log(n/\varepsilon))$ via binary search in the interval $[\delta_{\text{start}}/(2n), \delta_{\text{start}}]$, see Theorem 11. We thus focus on the approximate decider for the remainder of this section.

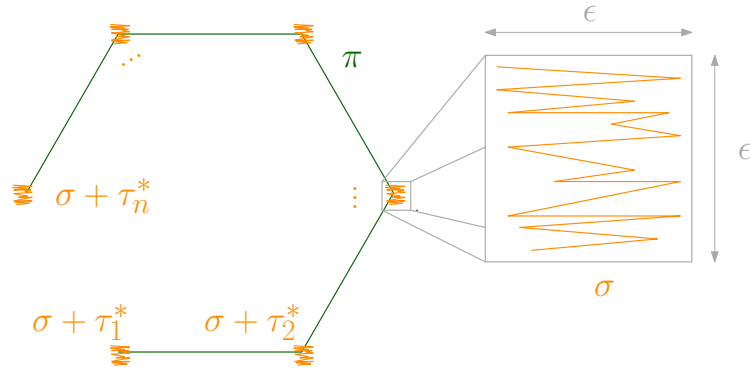
A simple $\mathcal{O}(n^4/\varepsilon^2)$ solution. Let B be the square of side length 2δ centered at $\tau_{\text{start}} = \pi_1 - \sigma_1$. To approximately decide whether $d_{\text{DTW}}(\pi, \sigma) \leq \delta$, we only need to consider translations in B , as any other translation τ incurs a DTW distance larger than δ by $d_{\text{DTW}}(\pi, \sigma + \tau) \geq \|\pi_1 - (\sigma_1 + \tau)\| = \|\tau_{\text{start}} - \tau\| > \delta$. Note that we can discretize this bounding box by a set Q of $\mathcal{O}((n/\varepsilon)^2)$ translations such that for each translation $\tau^* \in B$, there is a close translation $\tau \in Q$ with $\|\tau^* - \tau\| \leq \frac{\varepsilon\delta}{2n}$. Thus, if there is a translation τ^* with $d_{\text{DTW}}(\pi, \sigma + \tau^*) \leq \delta$, then by $(2n - 1)$ -Lipschitzness (Lemma 3), there is a translation $\tau \in Q$ with $d_{\text{DTW}}(\pi, \sigma + \tau) \leq d_{\text{DTW}}(\pi, \sigma + \tau^*) + (2n - 1) \cdot \|\tau^* - \tau\| \leq (1 + \varepsilon)\delta$. Consequently, by deciding $d_{\text{DTW}}(\pi, \sigma + \tau) \leq (1 + \varepsilon)\delta$ for all $\tau \in Q$ using the exact $\mathcal{O}(n^2)$ -time algorithm, we obtain an approximate decider with running time $T(n, \varepsilon) = \mathcal{O}(n^4/\varepsilon^2)$.

Note that the above arguments simplify the problem as follows: Find a set Q of translations such that if there is some *witness translation* τ^* , i.e., $d_{\text{DTW}}(\pi, \sigma + \tau^*) \leq \delta$, then there is some $\tau \in Q$ with $\|\tau - \tau^*\| \leq \frac{\varepsilon\delta}{2n}$. By computing $d_{\text{DTW}}(\pi, \sigma + \tau)$ for all $\tau \in Q$, we can then approximately decide whether $d_{\text{DTW}}^T(\pi, \sigma) \leq \delta$.

A more careful $\mathcal{O}(n^3/\varepsilon^2)$ solution. It turns out that we can significantly reduce the size of the set Q by analyzing the properties of good DTW traversals more closely. Consider a DTW traversal $((i_1, j_1), \dots, (i_t, j_t))$ of π and $\sigma + \tau^*$, with traversal cost $\sum_{\ell=1}^t \|\pi_{i_\ell} - (\sigma_{j_\ell} + \tau^*)\| \leq \delta$. Then, by a simple Markov argument, there can be at most $n/2$ pairs $\pi_{i_\ell}, \sigma_{j_\ell}$ with $\|\pi_{i_\ell} - (\sigma_{j_\ell} + \tau^*)\| \geq 2\delta/n$, since otherwise already these pairs would lead to a traversal cost of more than δ . Since the traversal has $t \geq n$ steps, it follows that there are at least $t - n/2 \geq n/2$ pairs $\pi_{i_\ell}, \sigma_{j_\ell}$ with $\|\pi_{i_\ell} - (\sigma_{j_\ell} + \tau^*)\| \leq 2\delta/n$. Since $\pi_\ell - (\sigma_\ell + \tau^*) = (\pi_\ell - \sigma_\ell) - \tau^*$, this yields an important restriction on τ^* :

For any τ^ such that π and $\sigma + \tau^*$ have DTW distance at most δ ,
there exist at least $n/2$ pairs π_i, σ_j with $\|(\pi_i - \sigma_j) - \tau^*\| \leq 2\delta/n$.*

This property immediately gives a simple randomized $\tilde{\mathcal{O}}(n^3/\varepsilon^2)$ algorithm: Simply draw a pair (i, j) uniformly at random from $[n]^2$ and test all translations τ given by $\mathcal{O}(1/\varepsilon^2)$ equally-spaced points in a $[-\frac{2\delta}{n}, \frac{2\delta}{n}]^2$ -box $C_{i,j}$ centered at $\pi_i - \sigma_j$. If there is some $\tau^* \in C_{i,j}$ with $d_{\text{DTW}}(\pi, \sigma + \tau^*) \leq \delta$, one of the checked translations τ achieves $d_{\text{DTW}}(\pi, \sigma + \tau) \leq (1 + \varepsilon)\delta$. By the above property, we have that $\tau^* \in C_{i,j}$ with probability at least $(n/2)/n^2 = 1/(2n)$. Thus, it suffices to repeat this process $\tilde{\mathcal{O}}(n)$ times to find a good translation with high probability, if one exists. This yields a total running time of $\tilde{\mathcal{O}}(n \cdot \frac{1}{\varepsilon^2} \cdot n^2) = \tilde{\mathcal{O}}(n^3/\varepsilon^2)$.



■ **Figure 2** If π is given by a regular n -gon and σ is a $3n$ -vertex curve in a small $[0, \varepsilon]^2$ area, where ε is small, then the DTW under translation distance can have $\Omega(n)$ local optima, each of which is near-optimal. These local optima correspond to translating σ towards each vertex of π .

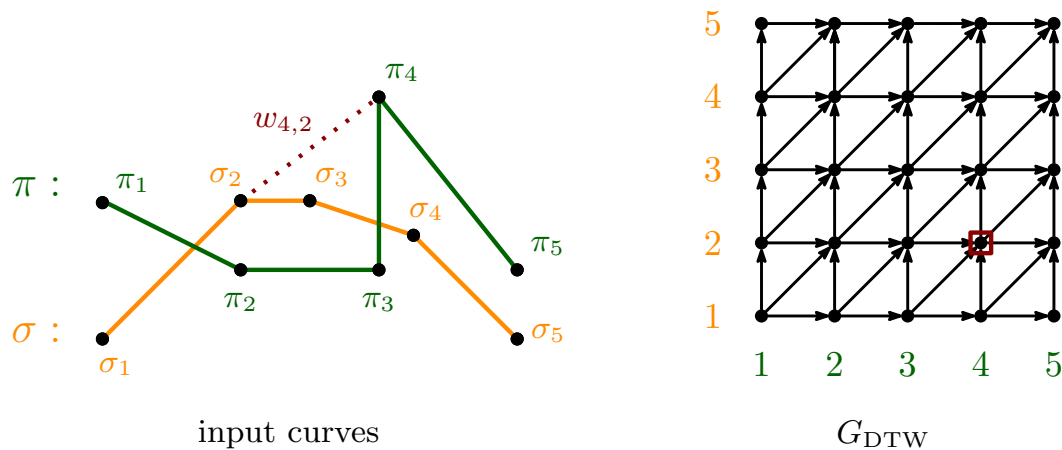
In order to leverage this property *deterministically*, define the multiset $P := \{\pi_i - \sigma_j \mid i, j \in [n]\}$ of n^2 points. Recall that B is the square of side length 2δ centered at $\tau_{\text{start}} = \pi_1 - \sigma_1$. We impose a grid on the bounding box B where each grid cell has side length $2\delta/n$. Consider a translation τ^* in some grid cell C such that π and $\sigma + \tau^*$ have DTW distance at most δ . Then there must be $n/2$ points $p \in P$ with $\|p - \tau^*\| \leq 2\delta/n$ – these points are distributed among C and at most 3 neighboring cells of C .² Thus, for any witness translation τ^* , there must be a neighboring (including itself) grid cell containing at least $n/8$ points from P – we call such a cell *dense*. Thus, we only need to check for translations that are inside a dense cell or neighboring a dense cell. Since $|P| = n^2$, there can be at most $|P|/(n/8) = 8n$ dense cells, resulting in $\mathcal{O}(n)$ cells to check for a good translation.

Since each grid cell has side length $\mathcal{O}(\delta/n)$, we can discretize each relevant cell C by $\mathcal{O}(1/\varepsilon^2)$ many translations Q_C such that if any $\tau^* \in C$ achieves $d_{\text{DTW}}(\pi, \sigma + \tau^*) \leq \delta$, then there is a $\tau \in Q_C$ with $\|\tau^* - \tau\| \leq \varepsilon\delta/(2n)$ and thus $d_{\text{DTW}}(\pi, \sigma + \tau) \leq (1 + \varepsilon)\delta$. Thus, by letting Q be the union of Q_C for all $\mathcal{O}(n)$ cells C that we need to check, we obtain $|Q| = \mathcal{O}(n/\varepsilon^2)$, significantly improving over the previous bound of $\mathcal{O}(n^2/\varepsilon^2)$. Computing the DTW distance for each translation in Q , we obtain a *deterministic* $\mathcal{O}(n^3/\varepsilon^2)$ -time algorithm.

Beating $\mathcal{O}(n^3/\varepsilon^2)$. Can we improve over the previous algorithm? A first idea would be to try to reduce the size of Q even further, below $\Theta(n \cdot \text{poly}(1/\varepsilon))$. However, there is evidence that this route is rather difficult: One can construct instances with $\Omega(n)$ many near-optimal local optima that are well-separated from each other, see Figure 2. It thus appears quite challenging to avoid a check of $\Omega(n)$ regions of translations.

A different route is to speed up the computation of DTW distances $d_{\text{DTW}}(\pi, \sigma + \tau)$ over all $\tau \in Q$, avoiding the naive time bound of $\mathcal{O}(|Q| \cdot n^2)$. Such approaches have been proven successful for related geometric optimization problems, such as Fréchet distance under translation [7, 13] or polygon placement [29]. Crucially, one needs to exploit that the $|Q|$ distance computations are related (for solving $|Q|$ independent instances, a conditional lower bound of $(|Q|n^2)^{1-o(1)}$ can be shown based on the quadratic-time hardness for DTW [1, 11]). To this end, we open up the black-box $\mathcal{O}(n^2)$ -algorithm for DTW.

² Here, we say that two cells are neighboring if they share a common vertex.



■ **Figure 3** An example of two curves π, σ and their dynamic time warping graph G_{DTW} .

Given $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_n)$, let G_{DTW} denote the node-weighted directed grid graph with vertex set $V = \{(i, j) \mid i, j \in [n]\}$ and edge set E consisting of horizontal edges from (i, j) to $(i+1, j)$, vertical edges from (i, j) to $(i, j+1)$ and diagonal edges from (i, j) to $(i+1, j+1)$. Each node (i, j) receives the weight $w_{i,j} = \|\pi_i - \sigma_j\|$. Then it is not difficult to see that $d_{DTW}(\pi, \sigma)$ is equal to the distance from $(1, 1)$ to (n, n) in G_{DTW} . As such, we can exploit algorithmic results on maintaining shortest paths in weighted planar digraphs under weight updates (here, one usually considers edge-weighted graphs, which subsumes the node-weighted setting). Unfortunately, when translating σ by τ , $\Omega(n^2)$ weights may change in G_{DTW} so that even constant-time updates would lead to an $\Omega(n^3/\varepsilon^2)$ time solution. In contrast, work on the Fréchet distance under translation [7, 13] considers translations in an order that incurs only $\mathcal{O}(1)$ updates per translation.

Surprisingly, one can indeed reduce the number of weight updates below $\mathcal{O}(n^2)$ when we resort to *approximating* each weight $w_{i,j}$ by an estimate $\|\pi_i - (\sigma_j + \tau)\|/(1 + \varepsilon) \leq w_{i,j} \leq (1 + \varepsilon) \cdot \|\pi_i - (\sigma_j + \tau)\|$. Specifically, we show how to traverse the $\mathcal{O}(n/\varepsilon^2)$ translations in Q in an order specified by a *space-filling curve* such that we only need to update $\tilde{\mathcal{O}}(n^2/\varepsilon^2)$ weights in total to maintain approximate weights. This statement and its analysis is one of the most interesting technical contributions of this paper and is proven in Section 3.2. It remains to report the shortest distance from $(1, 1)$ to (n, n) in the directed grid graph G_{DTW} for $\mathcal{O}(n/\varepsilon^2)$ queries and $\tilde{\mathcal{O}}(n^2/\varepsilon^2)$ weight updates. For this task, we use the data structure due to Das et al. [16] whose parameters can be set to give query time $\tilde{\mathcal{O}}(N^{3/4})$ and update time $\tilde{\mathcal{O}}(N^{1/4})$ for weighted planar digraphs with N vertices. Since $N = n^2$, we obtain a total running time of $\tilde{\mathcal{O}}(n^{2.5}/\varepsilon^2)$, which improves polynomially over the previous $\tilde{\mathcal{O}}(n^3/\varepsilon^2)$ solution. We believe that our approach of maintaining approximate weights efficiently using a space-filling curve traversal may turn out useful for further improvements in similar contexts of geometric optimization problems.

2 Preliminaries & notation

To denote index sets we use the notation $[n] := \{1, \dots, n\}$. Let $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ and $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_m)$ be two sequences of points in \mathbb{R}^d . We assume $n \geq m$ without loss of generality. To define the Dynamic Time Warping distance (DTW), we first introduce traversals. A sequence of index pairs $T = ((i_1, j_1), (i_2, j_2), \dots, (i_L, j_L))$ is a traversal of

20:8 (Approximation) Algorithms for Dynamic Time Warping Under Translation

two curves of complexity n and m if $(i_1, j_1) = (1, 1)$, $(i_L, j_L) = (n, m)$, and $(i_{\ell+1}, j_{\ell+1}) \in \{(i_\ell + 1, j_\ell), (i_\ell, j_\ell + 1), (i_\ell + 1, j_\ell + 1)\}$ for each $\ell \in [L - 1]$. We call L the number of steps of the traversal T . Let $\mathcal{T}_{n,m}$ be the set of all traversals of curves of length n and m . The Dynamic Time Warping distance between π and σ is then defined as

$$d_{\text{DTW}}(\pi, \sigma) := \min_{T \in \mathcal{T}} \sum_{(i,j) \in T} d(\pi_i, \sigma_j),$$

where for the metric $d(\cdot, \cdot)$, we use the L_p -norm $d(x, y) = \|x - y\|_p$ throughout this paper. In the remainder, we omit the p as it is either clear from the context, or the statement holds for all $p \in [1, \infty)$. Furthermore, we often use bounds on the number of steps of the traversal. To that end, note that for $m \leq n$, any traversal in $\mathcal{T}_{n,m}$ consists of at least n and at most $n + m - 1$ steps.

For a sequence $\pi = (\pi_1, \dots, \pi_n)$ with $\pi_i \in \mathbb{R}^d$ and a translation $\tau \in \mathbb{R}^d$, we define the translated sequence as $\pi + \tau := (\pi_1 + \tau, \pi_2 + \tau, \dots, \pi_n + \tau)$. Dynamic Time Warping Under Translation is then defined as $d_{\text{DTW}}^T(\pi, \sigma) := \min_{\tau \in \mathbb{R}^d} d_{\text{DTW}}(\pi, \sigma + \tau)$. Recall that a function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ is called L -Lipschitz (with respect to norm $\|\cdot\|$) if for any $\tau, \tau' \in \mathbb{R}^d$ we have $|f(\tau) - f(\tau')| \leq L \cdot \|\tau - \tau'\|$. We prove the following lemma in the full version.

► **Lemma 3.** $d_{\text{DTW}}(\pi, \sigma + \tau)$ is $(n + m - 1)$ -Lipschitz in τ .

The following lemma gives a simple $(n + m)$ -approximation for DTW under translation and is a straightforward adaption of a corresponding 2-approximation for the Fréchet distance under translation [12, Observation 2]. Note that one can create simple examples where this approximation ratio is almost tight. Again, we defer the proof to the full version.

► **Lemma 4.** Let $\tau_{\text{start}} = \pi_1 - \sigma_1$. Then $d_{\text{DTW}}(\pi, \sigma + \tau_{\text{start}}) \leq (n + m) \cdot d_{\text{DTW}}^T(\pi, \sigma)$.

As discussed in Section 1, DTW corresponds to a grid graph problem. We now formally define this. Given a DTW instance with curves $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_m)$, we define a directed graph $G_{\text{DTW}} = (V, E, w)$ on a node-weighted grid (including certain diagonals) with node set $V := \{(i, j) \mid i \in [n], j \in [m]\}$, edge set

$$E := \{((i, j), (i + 1, j)) \mid i \in [n - 1], j \in [m]\} \cup \{((i, j), (i, j + 1)) \mid i \in [n], j \in [m - 1]\} \\ \cup \{((i, j), (i + 1, j + 1)) \mid i \in [n - 1], j \in [m - 1]\},$$

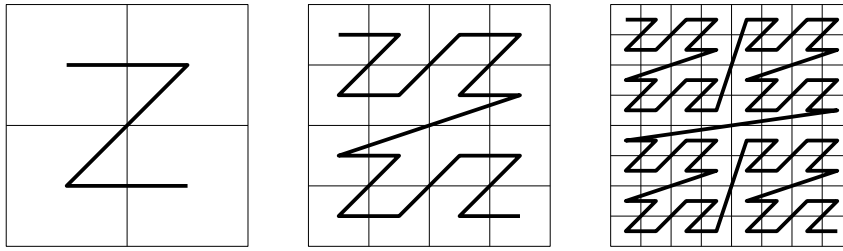
and weights $w: V \rightarrow \mathbb{R}$ with $w((i, j)) := \|\pi_i - \sigma_j\|$. To simplify notation, we write $w_{i,j}$ instead of $w((i, j))$ to denote the weight of node (i, j) . Note that finding a shortest path in this graph from $(1, 1)$ to (n, m) is equivalent to finding the minimum cost traversal.

In order to define the order of the updates and queries in the dynamic graph problem that we introduce in Section 3, we use a space-filling curve on a grid. Let

$$\mathcal{G}_R := \{i \cdot R \mid i \in \mathbb{Z}\} \times \{j \cdot R \mid j \in \mathbb{Z}\}.$$

be an infinite grid with resolution $R \in \mathbb{R}$. For our purpose, a space-filling curve is a hierarchical traversal of a finite grid: we partition this grid into four parts and, in some fixed order of the parts, recursively traverse each subgrid exhaustively before traversing the next one. More precisely, we define the curve on the $2^k \times 2^k$ grid $C_{0,0}^k := \mathcal{G}_R \cap [0, (2^k - 1)R]^2$ for some R , and we recursively split $C_{i,j}^\ell$ into the boxes $C_{2i,2j}^{\ell-1}, C_{2i,2j+1}^{\ell-1}, C_{2i+1,2j}^{\ell-1}, C_{2i+1,2j+1}^{\ell-1}$ until they only contain a single grid point, i.e., until $\ell = 0$. This leads to the following definition:

$$C_{i,j}^\ell := \{(i2^\ell + s)R \mid s \in \{0, \dots, 2^\ell - 1\}\} \times \{(j2^\ell + s)R \mid s \in \{0, \dots, 2^\ell - 1\}\}.$$



■ **Figure 4** The traversal of the z-curve for $k = 1$, $k = 2$, and $k = 3$.

For each cell $C_{i,j}^\ell$ with $\ell > 0$, the space-filling curve then traverses the points of the children in a way such that for each child all points are traversed in a continuous piece. For example, we first traverse all points of $C_{2i,2j+1}^{\ell-1}$, then $C_{2i+1,2j+1}^{\ell-1}$, then $C_{2i,2j}^{\ell-1}$, and finally $C_{2i+1,2j}^{\ell-1}$. Recursively applying this leads to a sequence $z_1, \dots, z_{2^{2k}}$ of all points in $\mathcal{G}_R \cap [0, (2^k - 1)R]^2$. This sequence is called the z-curve, see Figure 4. (However, any other order to traverse the children also works for our purpose.)

To argue about the space-filling curve traversals, it is sometimes useful to view the grid of points \mathcal{G}_R equivalently as a grid of cells, i.e., as a set of squares partitioning \mathbb{R}^2 . To switch between these views, build the Voronoi diagram of the point grid to obtain the cell grid, and conversely, use the center of each cell to obtain the point grid. We will freely use whichever view is most convenient in any context.

3 Approximating DTW under translation in L_p

In this section we present an $\tilde{O}(n^{2.5}/\varepsilon^2)$ algorithm for the problem of $(1 + \varepsilon)$ -approximating DTW under translation in the Euclidean plane. The algorithm that we present consists of two parts. First, we reduce to a dynamic shortest path problem on a grid graph. Second, we show that with the resulting number of updates and queries, we can use an existing dynamic graph algorithm to then obtain a subcubic algorithm for the problem at hand.

Recall that we consider the approximate decision problem: Given sequences $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_m)$ with $\pi_i, \sigma_j \in \mathbb{R}^2$, a distance $\delta \in \mathbb{R}$, and an approximation parameter $\varepsilon > 0$, either decide that $d_{\text{DTW}}^T(\pi, \sigma) \leq (1 + \varepsilon)\delta$ or that $d_{\text{DTW}}^T(\pi, \sigma) > \delta$. Recall that we assume $n \geq m$. We first present a basic cubic algorithm that already captures some important properties of the subcubic algorithm that we subsequently present.

3.1 Cubic algorithm

We now present the cubic algorithm that was already outlined in Section 1.2. First, if $\delta = 0$, we make a precise decision by testing for $d_{\text{DTW}}(\pi, \sigma + \tau_{\text{start}}) = 0$ with $\tau_{\text{start}} = \pi_1 - \sigma_1$. To facilitate the presentation, we furthermore assume that ε is given such that $\frac{n}{\varepsilon} = 2^k$ for some $k \in \mathbb{N}$. We can easily achieve this by rounding the input ε down to the largest value that fulfils this constraint, which changes the value of ε by at most a factor of 2.

In another preprocessing step, we round the coordinates of the points of π and σ to the closest multiple of $\frac{\delta}{4n}\varepsilon$. This is feasible as it changes the DTW distance by less than

$$(n + m) \cdot \frac{\delta}{4n}\varepsilon \leq \delta \frac{\varepsilon}{2}.$$

The multiset of translations from any point in σ to any point in π is then defined as

$$P := \{\pi_i - \sigma_j \mid i \in [n] \text{ and } j \in [m]\}.$$

20:10 (Approximation) Algorithms for Dynamic Time Warping Under Translation

Note that by construction also all coordinates of all points in P are multiples of $\frac{\delta}{4n}\varepsilon$. Furthermore, as P is a multiset, we have $|P| = nm$. We now define a set of boxes that enables us to find dense regions. Consider the square $B := [-\delta, \delta]^2 + \tau_{\text{start}}$. Partition B into n^2 boxes B_1, \dots, B_{n^2} of size $\frac{2\delta}{n} \times \frac{2\delta}{n}$ (note that their boundaries might intersect). We now formally define the notion of a dense box already introduced intuitively in Section 1.2.

► **Definition 5** (Dense Box). *A box B_i is dense if at least $\frac{n}{18}$ points of P are contained in B_i .*

As $|P| = nm$, we obtain the following observation:

► **Observation 6.** *There are at most $18m$ dense boxes.*

Note that we can find the dense boxes in time $\tilde{\mathcal{O}}(|P|)$ by associating each point with the tuple of indices in $[n] \times [n]$ of its containing box and then sorting these tuples. Now, let $N(B_i)$ be the neighborhood of a box B_i , i.e., $N(B_i) := \{B_j \mid B_j \cap B_i \neq \emptyset\}$. Note that $B_i \in N(B_i)$, so each box has (up to) 9 neighbors. The crucial property of dense boxes is that any witness translation τ with $d_{\text{DTW}}(\pi, \sigma + \tau) \leq \delta$ has to be in the neighborhood of a dense box:

► **Lemma 7.** *If $d_{\text{DTW}}^T(\pi, \sigma) \leq \delta$, then there exists a dense box B_j , a neighbor $B_i \in N(B_j)$, and a $\tau \in B_i$ such that $d_{\text{DTW}}(\pi, \sigma + \tau) \leq \delta$.*

The proof is deferred to the full version.

As we have to approximately decide whether $d_{\text{DTW}}(\pi, \sigma + \tau) \leq \delta$ for any τ that is neighboring a dense box, we intersect each of these boxes with an $8\varepsilon \times 8\varepsilon$ grid and this gives us the set of points Q that we have to evaluate. More precisely, let $\mathcal{G} := \mathcal{G}_{\frac{\delta}{4n}\varepsilon} \cap B$, where again $B = [-\delta, \delta]^2 + \tau_{\text{start}}$. Note that all points of \mathcal{G} are still integer multiples of $\frac{\delta}{4n}\varepsilon$. We now define our set of evaluation points to be

$$Q := \{\mathcal{G} \cap B_i \mid B_i \in N(B_j) \text{ for some dense box } B_j\}.$$

Note that from Observation 6 and the bound $|\mathcal{G} \cap B_i| \in \mathcal{O}(\frac{1}{\varepsilon^2})$, it follows that $|Q| \in \mathcal{O}(\frac{m}{\varepsilon^2})$.

Computing $d_{\text{DTW}}(\pi, \sigma + q)$ for each $q \in Q$ suffices to implement an approximate decider. Indeed, if for some $q \in Q$ we find $d_{\text{DTW}}(\pi, \sigma + q) \leq (1 + \varepsilon)\delta$, then we conclude that $d_{\text{DTW}}^T(\pi, \sigma) \leq (1 + \varepsilon)\delta$. Otherwise, if $d_{\text{DTW}}(\pi, \sigma + q) > (1 + \varepsilon)\delta$ for all $q \in Q$, then we conclude that $d_{\text{DTW}}^T(\pi, \sigma) > \delta$, by the following lemma proven in the full version.

► **Lemma 8** (Correctness). *If $d_{\text{DTW}}(\pi, \sigma + q) > (1 + \varepsilon)\delta$ for all $q \in Q$, then $d_{\text{DTW}}^T(\pi, \sigma) > \delta$.*

If we just evaluate each point in Q naively, then the running time is $\mathcal{O}(nm^2 (\frac{1}{\varepsilon})^2)$, as there are $\mathcal{O}(m)$ dense cells, each of them with $(\frac{1}{\varepsilon})^2$ grid points, and each DTW evaluation takes time $\mathcal{O}(nm)$. In the next section, instead of naively recomputing DTW for each translation, we dynamically update the DTW graph weights and then query for the shortest path.

3.2 Reduction to dynamic graph problem

Now we present the first step in solving DTW under translation in subcubic time. To this end, we transform our problem into a dynamic shortest path problem on a grid graph.

Dynamic graph problem

Recall that computing DTW for a fixed translation is a shortest path problem on a grid graph, see Section 2. More precisely, in the grid graph with node weights $w_{i,j} = \|\pi_i - (\sigma_j + q)\|$ the shortest path distance from $(1, 1)$ to (n, m) is equal to $d_{\text{DTW}}(\pi, \sigma + q)$. However, as we only want to compute a $(1 + \varepsilon)$ -approximation, we can relax the condition on the node weights to:

$$\frac{\|\pi_i - (\sigma_j + q)\|}{(1 + \varepsilon)} \leq w_{i,j} \leq (1 + \varepsilon)\|\pi_i - (\sigma_j + q)\|. \quad (1)$$

Observe that for such node weights the shortest path distance from $(1, 1)$ to (n, m) is equal to $d_{\text{DTW}}(\pi, \sigma + q)$ up to a factor $(1 + \varepsilon)$.

We choose the same set of query translations Q as in Section 3.1. We iterate over all $q \in Q$, and for each q we first update the node weights in the grid graph in order to satisfy (1) and then we query the shortest path distance from $(1, 1)$ to (n, m) in the grid graph, obtaining a $(1 + \varepsilon)$ -approximation of $d_{\text{DTW}}(\pi, \sigma + q)$. As in Section 3.1, this yields a $(1 + \mathcal{O}(\varepsilon))$ -approximation of $d_{\text{DTW}}^T(\pi, \sigma)$ (and after scaling ε this becomes a $(1 + \varepsilon)$ -approximation). Note that we did not fix the ordering of the query translations $q \in Q$ yet. In the following, we first fix this ordering, and then argue that our ordering guarantees that the total number of node weight updates is small, and furthermore we can efficiently determine which node weight updates have to be performed.

Query ordering

Consider the z-curve over the grid $\mathcal{G} = \mathcal{G}_{\frac{\delta}{4n}\varepsilon} \cap [-2\delta, 2\delta]^2 + \tau_{\text{start}}$. Note that this z-curve has depth $\log_2(16\frac{n}{\varepsilon})$, and recall that $\frac{n}{\varepsilon}$ is a power of 2. The points of Q lie on the z-curve, as π, σ are rounded and the grid has resolution $\frac{\delta}{4n}\varepsilon$. Thus, the z-curve induces an ordering of Q , and this is the ordering that we choose.

Updates

We now describe how we determine the node weight updates in the dynamic grid graph problem, to ensure that when we run the shortest path query corresponding to $q \in Q$ the node weights satisfy (1). We first argue why the total number of node weight updates is small, and subsequently discuss how to compute the sequence of node weight updates.

► **Lemma 9.** *Consider the sequence of points $\tau_1, \dots, \tau_{(16n/\varepsilon)^2}$ given by the z-curve on the grid \mathcal{G} , and fix i, j . Using only $\mathcal{O}(\frac{1}{\varepsilon^2} \log \frac{n}{\varepsilon})$ updates to the node weight $w_{i,j}$, we can maintain $w_{i,j}$ as a $(1 + \varepsilon)$ -approximation of the distance $\|\pi_i - (\sigma_j + \tau_k)\|$ while iterating over $k = 1, \dots, (16n/\varepsilon)^2$.*

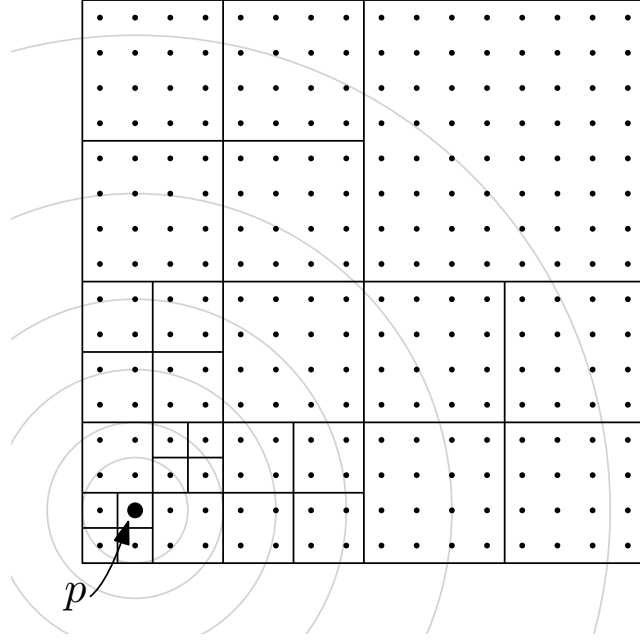
Proof. Note that for $p := \pi_i - \sigma_j$ we have $\|p - \tau\| = \|\pi_i - (\sigma_j + \tau)\|$, and thus the distance that we want to maintain is $\|p - \tau_k\|$. Additionally, note that p lies on the z-curve as the coordinates of the curves are rounded to $\frac{\delta}{4n}\varepsilon$ and, furthermore, if p is further than δ from $\tau_{\text{start}} = \pi_1 - \sigma_1$, then using the pair i, j in the traversal would incur a distance of more than δ for all $q \in Q$ and thus we can just set $w_{i,j} = \infty$.

Now, consider the following process on the recursive definition of the z-curve: Starting with the root $C_{0,0}$, recursively explore all children of each cell. Stop the process at a cell C if there is an $\ell \in \mathbb{Z}$ such that all points $\tau \in C$ have a distance

$$(1 + \varepsilon)^{\ell-1} \leq \|p - \tau\| \leq (1 + \varepsilon)^{\ell+1}.$$

In this case, for all points in C the value $(1 + \varepsilon)^\ell$ is a valid $(1 + \varepsilon)$ -approximation for the distance to p , so we associate the distance $(1 + \varepsilon)^\ell$ with C .

Note that the process is well-defined, since at the lowest level of the recursion a cell contains only a single point of \mathcal{G} , and thus at the latest on this level the process will stop. The process partitions the grid \mathcal{G} into cells C_1, \dots, C_t , which are exhaustively explored in this order by the z-curve, see Figure 5 for an illustration. In particular, the value of t is an upper bound on the number of updates needed to approximately maintain $\|p - \tau_k\|$ while iterating over all points in \mathcal{G} in z-order. We next bound the diameter of the cells for a specific associated distance, to subsequently show that this induces a small number of



■ **Figure 5** The partition of cells induced by the point p on the recursively defined cells of the z-curve such that for each cell a single update suffices to ensure a $(1 + \varepsilon)$ -approximation on the distance to p . Note that the cells become larger when further away from p .

cells in the partition. To this end, consider a specific cell C_r and its associated distance $(1 + \varepsilon)^\ell$. As we continued exploring the children of the parent cell C of C_r , there have to be two points $z_1, z_2 \in C$ such that either $\|z_1 - p\| < (1 + \varepsilon)^{\ell-1}$ and $\|z_2 - p\| > (1 + \varepsilon)^\ell$, or $\|z_1 - p\| < (1 + \varepsilon)^\ell$ and $\|z_2 - p\| > (1 + \varepsilon)^{\ell+1}$. By triangle inequality, C has diameter at least

$$\|z_1 - z_2\| \geq \|z_2 - p\| - \|z_1 - p\| > (1 + \varepsilon)^\ell - (1 + \varepsilon)^{\ell-1} = (1 + \varepsilon)^{\ell-1}((1 + \varepsilon) - 1) = (1 + \varepsilon)^{\ell-1}\varepsilon.$$

In the recursive definition of the z-curve, the diameter of a cell decreases at most by a constant factor from parent to child if the child is not a single point. Thus C_j has diameter $\Omega((1 + \varepsilon)^\ell \varepsilon)$ if $|C_j| > 1$. If $|C_j| = 1$, then the Voronoi cell of C_j of the Voronoi diagram of \mathcal{G} has diameter $\Omega((1 + \varepsilon)^\ell \varepsilon)$. Thus, for both cases it holds that there is a square with area $\Omega((1 + \varepsilon)^{2\ell} \varepsilon^2)$ that only contains points from C_j but no other cell $C_{j'}, j' \neq j$.

Recall that the area of a ball of radius R in the L_ρ -norm is equal to $\nu_\rho R^2$, where ν_ρ depends only on the L_ρ -norm and is thus a constant for our purpose. Hence, the area of all points between distance $(1 + \varepsilon)^{\ell-1}$ and $(1 + \varepsilon)^{\ell+1}$ from p is equal to

$$\nu_\rho(1 + \varepsilon)^{2(\ell+1)} - \nu_\rho(1 + \varepsilon)^{2(\ell-1)} = \nu_\rho(1 + \varepsilon)^{2(\ell-1)}((1 + \varepsilon)^4 - 1) = \mathcal{O}((1 + \varepsilon)^{2\ell} \varepsilon).$$

Thus, there can be at most

$$\frac{\mathcal{O}((1 + \varepsilon)^{2\ell} \varepsilon)}{\Omega((1 + \varepsilon)^{2\ell} \varepsilon^2)} = \mathcal{O}\left(\frac{1}{\varepsilon}\right)$$

cells associated with distance $(1 + \varepsilon)^\ell$. Finally, there are at most $\mathcal{O}(\log_{1+\varepsilon} \frac{n}{\varepsilon}) = \mathcal{O}(\frac{1}{\varepsilon} \log \frac{n}{\varepsilon})$ different associated distances, as the minimum non-zero distance is $\Omega(\frac{\delta}{n} \varepsilon)$ and the largest distance is $\mathcal{O}(\delta)$. Consequently, the total number of updates can be bounded by $\mathcal{O}(\frac{1}{\varepsilon^2} \log \frac{n}{\varepsilon})$.

◀

We now discuss how we explicitly compute the updates. Note that explicitly checking for updates in each node of the traversal of the z-curve is prohibitive. Thus, we have to devise a non-naive way of computing the updates. Indeed, Lemma 9 can be turned into an algorithm.

► **Lemma 10.** *The updates in Lemma 9 can explicitly be computed in time $\mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \log \frac{n}{\varepsilon}\right)$.*

Proof. Lemma 9 already is constructive, as we associated updates to cells and thereby we can simply perform these updates at the first point of such cells. It therefore only remains to bound the running time of all steps. The running time for exploring the z-curve tree is dominated by the number of cells in the partition, i.e., by the number of updates, multiplied with the running time of deciding whether to explore further or not. If the point $p = \pi_i - \sigma_j$ of Lemma 9 is contained in the currently considered cell, then we have to continue exploring. Otherwise, we can check if all points lie in a $(1 + \varepsilon)^{\ell-1}$ to $(1 + \varepsilon)^{\ell+1}$ distance window for any $\ell \in \mathbb{Z}$ by computing the distance to the closest and furthest point in the cell from p . All of the above steps can be done in $\mathcal{O}(1)$ time. Finally, note that no sorting of the updates is necessary, as exploring the z-curve tree via depth-first search in the order of the z-curve already constructs the updates in sorted order. Hence, the running time of explicitly computing the updates is dominated by the number of updates itself. ◀

We can directly use Lemma 10 to compute the updates for all node weights $w_{i,j}$. However, computing them separately would additionally incur the cost of merging them into a sorted order. We can avoid this sorting step by constructing the updates for all node weights $w_{i,j}$ in parallel using a single DFS on the z-order tree. During the DFS, we maintain a set E of pairs (i, j) for which recursing further is necessary; in the top cell, this is set to $[n] \times [m]$. Then for each cell in the DFS, we need to decide for each pair $(i, j) \in E$ whether a single weight $w_{i,j} = (1 + \varepsilon)^\ell$ suffices to approximate the distance in this cell, which in total takes time $\mathcal{O}(|E|)$. If for $(i, j) \in E$ this is the case, then we add an update of $w_{i,j}$ to $(1 + \varepsilon)^\ell$ for the first point τ in this cell, and remove (i, j) from E for the recursive calls that explore the children of this cell. (We add back (i, j) after the exploration.) This process creates the updates in order and thus we do not have to sort them in a postprocessing step. It follows that the updates for all node weights $w_{i,j}$ can explicitly be computed in time $\mathcal{O}\left(nm \frac{1}{\varepsilon^2} \cdot \log \frac{n}{\varepsilon}\right)$.

Main theorem

Finally, we obtain our main theorem.

► **Theorem 11.** *Assume a data structure for approximate shortest paths in a directed grid graph with N vertices and fixed vertices s, t , supporting updates of an edge weight in time $U(N)$ and $(1 + \varepsilon)$ -approximate s - t -distance queries in time $Q(N)$. We can $(1 + \varepsilon)$ -approximate DTW under translation in L_p -norm in time $\mathcal{O}\left(U(nm) \frac{nm}{\varepsilon^2} \log^2 \frac{n}{\varepsilon} + Q(nm) \frac{m}{\varepsilon^2} \log \frac{n}{\varepsilon}\right)$.*

The proof is deferred to the full version; it follows easily by combining the above arguments.

3.3 Solving the dynamic graph problem

Consider the data structure assumed in Theorem 11 for maintaining shortest paths in a directed grid graph. Das et al. [16] obtain a trade-off of update time $U(N) = \tilde{\mathcal{O}}(N^r)$ and query time $Q(N) = \tilde{\mathcal{O}}(N^{1-r})$ even for *exact* distance queries in directed planar graphs where $r \in [0, \frac{1}{2}]$ is an adjustable parameter, all updates are given in advance, and all edge-weights are non-negative (both are the case in our setting). The aforementioned result improves

bounds due to Fakcharoenphol and Rao [21] and Klein [28], also see [27, 22], by considering the offline setting. For tight conditional lower bounds for the offline setting, we refer to [2]. By setting r such that $N^r = \sqrt{m}$ (which satisfies $r \in [0, \frac{1}{2}]$), we obtain the following corollary.

► **Corollary 12.** *We can $(1+\varepsilon)$ -approximate DTW under translation in \mathbb{R}^2 under the L_p -norm in time $\tilde{O}(nm^{1.5}/\varepsilon^2)$.*

Note that for $n = m$ this becomes $\tilde{O}(n^{2.5}/\varepsilon^2)$. It is straightforward to generalize our algorithm to \mathbb{R}^d for constant d . To this end, we have to replace the 2-dimensional ε -grid and the 2-dimensional space-filling curve by their d -dimensional counterparts and adapt the analysis accordingly. The running time then merely increases with respect to the dependency on ε . See the full version for the proof of the following corollary.

► **Corollary 13.** *We can $(1+\varepsilon)$ -approximate DTW under translation in \mathbb{R}^d under the L_p -norm with $d \in \mathcal{O}(1)$ in time $\tilde{O}(nm^{1.5}/\varepsilon^d)$.*

4 Conclusion and open problems

We give the first rigorous algorithms for Dynamic Time Warping under translation, specifically an exact $\mathcal{O}(n^{2(d+1)})$ -time algorithm for the L_1 norm in \mathbb{R}^d , as well as a $(1+\varepsilon)$ -approximate $\tilde{O}(n^{2.5}/\varepsilon^2)$ -time algorithm for the L_p -norm in \mathbb{R}^2 .

The most interesting open problem is to determine whether under the L_2 -norm, DTW under translation admits an $\tilde{O}(n^2 f(1/\varepsilon))$ -time approximation scheme. In fact, one might be able to improve over our $\tilde{O}(n^{2.5}/\varepsilon^2)$ -time algorithm via purely graph-theoretic improvements for dynamic shortest path algorithms in grid graphs, applying Theorem 11 as a black box. Specifically, we showed how to reduce $(1+\varepsilon)$ -approximate DTW under translation to (approximately) maintaining the s - t distance in a directed grid graph undergoing edge-weight updates. Our precise bound follows from plugging in a data structure due to Das et al. [16] that maintains all *exact* distances. In fact, compared to their setting, our target problem has several important restrictions that may help to design faster algorithms:

- Instead of exact distances, our application only requires a $(1+\varepsilon)$ -approximation.
- Our restriction to directed grid graphs might turn out significantly simpler than general planar digraphs.
- We only ever query the distance between a single source-sink pair.

Finally, if no further algorithmic improvements can be found, can we give improved conditional hardness results, going beyond our reduction from DTW for a fixed translation?

References

- 1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 59–78. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.14.
- 2 Amir Abboud and Søren Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 477–486. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.58.
- 3 Pankaj K. Agarwal, Kyle Fox, Jiangwei Pan, and Rex Ying. Approximating dynamic time warping and edit distance for a pair of point sequences. In Sándor P. Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, volume 51 of *LIPICs*, pages 6:1–6:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.SoCG.2016.6.

- 4 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5(1–2):78–99, 1995.
- 5 Helmut Alt, Christian Knauer, and Carola Wenk. Matching polygonal curves with respect to the Fréchet distance. In *Proc. 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS'01)*, pages 63–74, 2001.
- 6 Chandrajit L. Bajaj. The algebraic degree of geometric optimization problems. *Discret. Comput. Geom.*, 3:177–191, 1988. doi:10.1007/BF02187906.
- 7 Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. A faster algorithm for the discrete Fréchet distance under translation. *ArXiv preprint*, 2015. arXiv:1501.03724.
- 8 Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *Proc. 31st International Conf. Very Large Data Bases (VLDB'05)*, pages 853–864, 2005.
- 9 Milutin Brankovic, Kevin Buchin, Koen Klaren, André Nusser, Aleksandr Popov, and Sampson Wong. (k, l) -medians clustering of trajectories using continuous dynamic time warping. In Chang-Tien Lu, Fusheng Wang, Goce Trajcevski, Yan Huang, Shawn D. Newsam, and Li Xiong, editors, *SIGSPATIAL '20: 28th International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, November 3-6, 2020*, pages 99–110. ACM, 2020. doi:10.1145/3397536.3422245.
- 10 Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly sub-quadratic algorithms unless SETH fails. In *Proc. 55th Ann. IEEE Symposium on Foundations of Computer Science (FOCS'14)*, pages 661–670, 2014.
- 11 Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 79–97. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.15.
- 12 Karl Bringmann, Marvin Künnemann, and André Nusser. When Lipschitz walks your dog: Algorithm engineering of the discrete Fréchet distance under translation. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPICs*, pages 25:1–25:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ESA.2020.25.
- 13 Karl Bringmann, Marvin Künnemann, and André Nusser. Discrete Fréchet distance under translation: Conditional hardness and an improved algorithm. *ACM Trans. Algorithms*, 17(3):25:1–25:42, 2021. doi:10.1145/3460656.
- 14 Maike Buchin. *On the computability of the Fréchet distance between triangulated surfaces*. PhD thesis, Freie Universität Berlin, 2007. PhD Thesis.
- 15 Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In Daniel Wichs and Yishay Mansour, editors, *Proc. 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2016)*, pages 9–21. ACM, 2016. doi:10.1145/2897518.2897647.
- 16 Debarati Das, Maximilian Probst Gutenberg, and Christian Wulff-Nilsen. A near-optimal offline algorithm for dynamic all-pairs shortest paths in planar digraphs. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3482–3495, 2022. doi:10.1137/1.9781611977073.138.
- 17 Mark de Berg and Atlas F. Cook. Go with the flow: The direction-based Fréchet distance of polygonal curves. In Alberto Marchetti-Spaccamela and Michael Segal, editors, *Theory and Practice of Algorithms in (Computer) Systems - First International ICST Conference, TAPAS 2011, Rome, Italy, April 18-20, 2011. Proceedings*, volume 6595 of *Lecture Notes in Computer Science*, pages 81–91. Springer, 2011. doi:10.1007/978-3-642-19754-3_10.
- 18 Alon Efrat, Quanfu Fan, and Suresh Venkatasubramanian. Curve Matching, Time Warping, and Light Fields: New Algorithms for Computing Similarity between Curves. *Journal of Mathematical Imaging and Vision*, 27(3):203–216, April 2007. doi:10.1007/s10851-006-0647-0.

20:16 (Approximation) Algorithms for Dynamic Time Warping Under Translation

- 19 Alon Efrat, Piotr Indyk, and Suresh Venkatasubramanian. Pattern matching for sets of segments. In *Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, pages 295–304, 2001.
- 20 Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 1994.
- 21 Jittat Fakcharoenphol and Satish Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. *J. Comput. Syst. Sci.*, 72(5):868–889, 2006. doi:10.1016/j.jcss.2005.05.007.
- 22 Pawel Gawrychowski and Adam Karczmarz. Improved bounds for shortest paths in dense distance graphs. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPICs*, pages 61:1–61:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.ICALP.2018.61.
- 23 Omer Gold and Micha Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Trans. Algorithms*, 14(4):50:1–50:17, 2018. doi:10.1145/3230734.
- 24 Joachim Gudmundsson, André van Renssen, Zeinab Saeidi, and Sampson Wong. Translation invariant Fréchet distance queries. *Algorithmica*, 83(11):3514–3533, 2021. doi:10.1007/s00453-021-00865-0.
- 25 Sarel Har-Peled. *Geometric approximation algorithms*, chapter Fréchet distance: How to walk your dog, pages 383–412. American Mathematical Society, 2017. Online chapter.
- 26 Minghui Jiang, Ying Xu, and Binhai Zhu. Protein structure–structure alignment with discrete Fréchet distance. *J. Bioinformatics and Computational Biology*, 6(01):51–64, 2008.
- 27 Haim Kaplan, Shay Mozes, Yahav Nussbaum, and Micha Sharir. Submatrix maximum queries in monge matrices and partial monge matrices, and their applications. *ACM Trans. Algorithms*, 13(2):26:1–26:42, 2017. doi:10.1145/3039873.
- 28 Philip N. Klein. Multiple-source shortest paths in planar graphs. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 146–155. SIAM, 2005. URL: <http://dl.acm.org/citation.cfm?id=1070432.1070454>.
- 29 Marvin Künnemann and André Nusser. Polygon placement revisited: (Degree of Freedom + 1)-SUM hardness and an improvement via offline dynamic rectangle union. In *Proc. 33rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*, 2022. To appear.
- 30 William Kuzmaul. Dynamic time warping in strongly subquadratic time: Algorithms for the low-distance regime and approximate evaluation. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 80:1–80:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ICALP.2019.80.
- 31 Anil Maheshwari, Jörg-Rüdiger Sack, and Christian Scheffer. Approximating the integral Fréchet distance. *Comput. Geom.*, 70-71:13–30, 2018. doi:10.1016/j.comgeo.2018.01.001.
- 32 Axel Mosig and Michael Clausen. Approximately matching polygonal curves with respect to the Fréchet distance. *Computational Geometry: Theory and Applications*, 30(2):113–127, 2005.
- 33 Mario E. Munich and Pietro Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*, pages 108–115. IEEE Computer Society, 1999. doi:10.1109/ICCV.1999.791205.
- 34 Mark H. Overmars and Chee-Keng Yap. New upper bounds in klee’s measure problem. *SIAM J. Comput.*, 20(6):1034–1045, 1991. doi:10.1137/0220065.
- 35 Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Elastic Translation Invariant Matching of Trajectories. *Machine Learning*, 58(2):301–334, February 2005. doi:10.1007/s10994-005-5830-9.

- 36 Carola Wenk. *Shape matching in higher dimensions*. PhD thesis, Freie Universität Berlin, 2003. PhD Thesis.
- 37 Carola Wenk, Randall Salas, and Dieter Pfoser. Addressing the need for map-matching speed: Localizing globalb curve-matching algorithms. In *18th International Conference on Scientific and Statistical Database Management, SSDBM 2006, 3-5 July 2006, Vienna, Austria, Proceedings*, pages 379–388. IEEE Computer Society, 2006. doi:10.1109/SSDBM.2006.11.
- 38 Rex Ying, Jiangwei Pan, Kyle Fox, and Pankaj K. Agarwal. A simple efficient approximation algorithm for dynamic time warping. In Siva Ravada, Mohammed Eunus Ali, Shawn D. Newsam, Matthias Renz, and Goce Trajcevski, editors, *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2016, Burlingame, California, USA, October 31 - November 3, 2016*, pages 21:1–21:10. ACM, 2016. doi:10.1145/2996913.2996954.
- 39 Yu Qiao and M. Yasuhara. Affine Invariant Dynamic Time Warping and its Application to Online Rotated Handwriting Recognition. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 905–908, August 2006. ISSN: 1051-4651. doi: 10.1109/ICPR.2006.228.

Towards Sub-Quadratic Diameter Computation in Geometric Intersection Graphs

Karl Bringmann ✉

Universität des Saarlandes, Saarbrücken, Germany

Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Sándor Kisfaludi-Bak ✉

Aalto University, Espoo, Finland

Marvin Künnemann ✉

Institute for Theoretical Studies, ETH Zürich, Switzerland

André Nusser ✉ 

BARC, University of Copenhagen, Denmark

Zahra Parsaeian ✉

Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Abstract

We initiate the study of diameter computation in geometric intersection graphs from the fine-grained complexity perspective. A geometric intersection graph is a graph whose vertices correspond to some shapes in d -dimensional Euclidean space, such as balls, segments, or hypercubes, and whose edges correspond to pairs of intersecting shapes. The diameter of a graph is the largest distance realized by a pair of vertices in the graph.

Computing the diameter in near-quadratic time is possible in several classes of intersection graphs [Chan and Skrepetos 2019], but it is not at all clear if these algorithms are optimal, especially since in the related class of planar graphs the diameter can be computed in $\tilde{O}(n^{5/3})$ time [Cabello 2019, Gawrychowski et al. 2021].

In this work we (conditionally) rule out sub-quadratic algorithms in several classes of intersection graphs, i.e., algorithms of running time $\mathcal{O}(n^{2-\delta})$ for some $\delta > 0$. In particular, there are no sub-quadratic algorithms already for fat objects in small dimensions: unit balls in \mathbb{R}^3 or congruent equilateral triangles in \mathbb{R}^2 . For unit segments and congruent equilateral triangles, we can even rule out strong sub-quadratic approximations already in \mathbb{R}^2 . It seems that the hardness of approximation may also depend on dimensionality: for axis-parallel unit hypercubes in \mathbb{R}^{12} , distinguishing between diameter 2 and 3 needs quadratic time (ruling out $(3/2-\varepsilon)$ -approximations), whereas for axis-parallel unit squares, we give an algorithm that distinguishes between diameter 2 and 3 in near-linear time.

Note that many of our lower bounds match the best known algorithms up to sub-polynomial factors. Ultimately, this fine-grained perspective may enable us to determine for which shapes we can have efficient algorithms and approximation schemes for diameter computation.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Hardness in P, Geometric Intersection Graph, Graph Diameter, Orthogonal Vectors, Hyperclique Detection

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.21

Related Version *Full Version*: <https://arxiv.org/abs/2203.03663>

Funding *Karl Bringmann*: This work is part of the project TIPEA that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 850979).

Sándor Kisfaludi-Bak: Part of this research was conducted while the author was at the Max Planck Institute for Informatics, and part of it while he was at the Institute for Theoretical Studies, ETH Zürich.



© Karl Bringmann, Sándor Kisfaludi-Bak, Marvin Künnemann, André Nusser, and Zahra Parsaeian;

licensed under Creative Commons License CC-BY 4.0

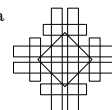
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 21; pp. 21:1–21:16

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Marvin Künnemann: Research supported by Dr. Max Rössler, by the Walter Haefner Foundation, and by the ETH Zürich Foundation. Part of this research was conducted while the author was at the Max Planck Institute for Informatics.

André Nusser: Part of this research was conducted while the author was at Saarbrücken Graduate School of Computer Science and Max Planck Institute for Informatics. The author is supported by the VILLUM Foundation grant 16582.

1 Introduction

The diameter of a simple graph $G = (V, E)$ is the largest distance realized by a pair of its vertices; formally, it is $\text{diam}(G) = \max_{u, v \in V} \text{dist}_G(u, v)$, where $\text{dist}_G(u, v)$ is the number of edges on a shortest path from u to v . It is one of the crucial parameters of a graph that can be computed in polynomial time. Geometric intersection graphs are the standard model for wireless communication networks [34], but more abstractly, they can be used to represent networks where the connection of nodes relies on proximity in some metric space. For a (slightly oversimplified) example, consider a set of devices in the plane capable of receiving and transmitting information in a range of radius 2. These devices form a communication network that is a unit disk graph. Indeed, two devices can communicate with each other if and only if their distance is at most 2, i.e., if the unit disks centered at the devices have a non-empty intersection. For our purposes, the underlying metric space will be d -dimensional Euclidean space (henceforth denoted by \mathbb{R}^d), and we will consider intersection graphs of common objects such as balls and segments. For a set F of objects in \mathbb{R}^d (that is, $F \subset 2^{\mathbb{R}^d}$), the corresponding intersection graph $G[F]$ has vertex set F and edge set $\{uv \mid u, v \in F, u \cap v \neq \emptyset\}$.

Computing the diameter in geometric intersection graphs is an important task: if the graph represents a communication network, then the diameter of the network can help estimate the time required to spread information in the network, as the information needs to go through up to $\text{diam}(G)$ links to reach its destination. In large networks, it is also indispensable to have near-linear time algorithms; it is therefore natural to study if a given class of geometric intersection graphs admits a near-linear time algorithm for exact or approximate diameter computation.

The extensive literature on diameter computation serves as a good starting point. The diameter of an n -vertex (unweighted) graph can be computed in $\mathcal{O}(n^\omega \log n)$ expected time, where $\omega < 2.37286$ is the exponent of matrix multiplication [41]. If the graph has m edges, then the diameter can also be computed in $\mathcal{O}(mn)$ time [42], which gives a near-quadratic running time of $\tilde{\mathcal{O}}(n^2)$ in case of sparse graphs, i.e., when $m = \tilde{\mathcal{O}}(n)$. In fact, these algorithms are capable of computing not only the diameter, but also all pairwise distances in a graph, known as the all pairs shortest paths problem.

On the negative side, we know that computing the diameter of a graph cannot be done in $\mathcal{O}(n^{2-\varepsilon})$ time under the Orthogonal Vectors Hypothesis¹ (OV); in fact, deciding if the diameter of a sparse graph is at most 2 or at least 3 requires $n^{2-o(1)}$ time under OV [40]², which rules out sub-quadratic $(3/2 - \varepsilon)$ -approximations for all $\varepsilon > 0$.

In special graph classes however it is possible to compute the diameter in sub-quadratic time. In planar graphs, an algorithm with running time $\mathcal{O}(n^2)$ is very easy: one can just run n breadth-first searches, each of which take linear time because the number of edges is $\mathcal{O}(n)$.

¹ See Section 2 for the definitions and some background on the hypotheses used in our lower bounds.

² More precisely, Roditty and Vassilevska-Williams [40] give a reduction from k -Dominating Set, which can be adapted to a reduction from OV as described in the beginning of Section 4.

It has been a long-standing open problem whether a truly sub-quadratic algorithm exists for diameter computation, until the breakthrough of Cabello [12], who used Voronoi diagrams in planar graphs. The technique was later improved by Gawrychowski *et al.* [29], who obtained a running time of $\tilde{O}(n^{5/3})$.

Certain geometric intersection graphs often behave similarly to planar graphs. The most widely studied classes, (unit) disk and ball graphs admit approximation schemes for maximum independent set, maximum dominating set, and several other problems [30, 31, 14], with techniques similar to planar graphs. Unlike planar graphs, geometric intersection graphs can have arbitrarily large cliques, but at least the maximum clique can be approximated efficiently [7]. In fact, planar graphs are special disk intersection graphs by the circle packing theorem [33]. When it comes to computing the diameter, the similarity with planar graphs is not so easy to see. Even getting near-quadratic diameter algorithms is non-trivial, as geometric intersection graphs can be arbitrarily dense.

Chan and Skrepetos [16] provide near-quadratic ($\tilde{O}(n^2)$) APSP algorithms for several graph classes, including disks, axis-parallel segments, and fat triangles in the plane, and cubes and boxes in constant-dimensional space. Unit disk graphs have a “weakly” sub-quadratic algorithm (that is poly-logarithmically faster than $\mathcal{O}(n^2)$) [15]. We are not aware of any $\mathcal{O}(n^{2-\epsilon})$ algorithms for computing the diameter in intersection graphs of any planar shape.

Further related work. While computing the diameter is known to require time $n^{2\pm o(1)}$ already on sparse graphs (assuming the OV Hypothesis), an extensive line of research including [3, 40, 19, 13, 5, 25, 9, 24, 37, 8, 23] studies the (non-)existence of faster approximation algorithms. On the positive side, this includes in particular a folklore 2-approximation in time $\tilde{O}(m)$ and a 3/2-approximation in time $\tilde{O}(m^{3/2})$ [3, 40, 19], both already for weighted digraphs. Remarkably, these algorithms can be shown to be tight: [40, 5] establish that the 3/2-approximation in time $\tilde{O}(m^{3/2})$ cannot be improved in either approximation guarantee or running time (assuming the k -OV Hypothesis), already for unweighted undirected graphs. The near-linear time 2-approximation is conditionally optimal as well: For unweighted directed graphs, this has been proven independently in [24, 37]. For unweighted undirected graphs, following further work [8], a resolution has been announced only very recently [23]. Thus, approximating the diameter in sparse graphs is quite well understood, including detailed insights into the full accuracy-time trade-off. In the context of our work, the challenge is to obtain a similar understanding for our setting of unweighted, undirected *geometric* graphs, which are non-sparse in general.

Note that for graph classes that are non-sparse, a natural question is whether diameter can be computed in $\mathcal{O}(m+n)$ time, i.e., linear time in the number of edges plus vertices. The question has been studied by several authors: using a variant of breadth-first search called *lexicographic breadth-first search*, one can find a vertex of very large eccentricity. In some classes, we now know that there is an $\mathcal{O}(m+n)$ algorithm for diameter: notably, this holds in interval graphs as well as {claw,asteroidal triple}-free graphs [27, 10]. In many other graph classes (such as chordal graphs and asteroidal-triple-free graphs) we can get approximations for the diameter that differ only by a small additive constant from the optimum [27, 26, 21]. See [22] for an overview on the connection of lexicographic BFS and diameter, and see [20] for a survey on lexicographic BFS.

Another related direction is to consider edge weighted graph classes. In some classes of geometric intersection graphs there is a natural weighting to consider: for example in ball graphs, it is customary to draw the graph edges with straight segments that connect the centers of the two adjacent disks. The edges then have a natural weighting by their Euclidean

length. This was considered for unit disk graphs in the plane by Gao and Zhang [28], who obtained a $(1 + \varepsilon)$ -approximation for DIAMETER in $\mathcal{O}(n^{3/2})$ time for any fixed $\varepsilon > 0$. A faster $(1 + \varepsilon)$ -approximation with running time $\mathcal{O}(n \log^2 n)$ for any fixed $\varepsilon > 0$ was given by Chan and Skrepetos [17]. Since the underlying graph is not changed by this weighting, it is natural to think that similar results should be possible also for unweighted unit disk graphs. It remains an open question whether the complexity of diameter computation is influenced by the presence of these Euclidean weights.

Our results. In this article, we show that most of the results of Chan and Skrepetos [16] cannot be significantly improved under standard complexity-theoretic assumptions, even if we are only interested in the diameter instead of all pairs shortest paths. In particular, we rule out sub-quadratic diameter algorithms for fat triangles and axis-aligned segments in the plane, as well as for unit cubes in \mathbb{R}^3 , leaving only their $\tilde{\mathcal{O}}(n^{7/3})$ algorithm for arbitrary segments in \mathbb{R}^2 as well as their $\tilde{\mathcal{O}}(n^2)$ algorithm for disks without a matching lower bound.

The DIAMETER problem has as input a set of geometric objects in \mathbb{R}^d and a number k ; the goal is to decide whether the diameter of the intersection graph of the objects is at most k . The DIAMETER- t problem is the same problem, but with k set to the constant number t . We show the following lower bounds.

- **Theorem 1.** *For all $\delta > 0$ there is no $\mathcal{O}(n^{2-\delta})$ time algorithm for*
 - *DIAMETER-3 in intersection graphs of unit segments in \mathbb{R}^2 under the OV Hypothesis.*
 - *DIAMETER-3 in intersection graphs of congruent equilateral triangles in \mathbb{R}^2 under the OV Hypothesis.*
 - *DIAMETER in intersection graphs of unit balls in \mathbb{R}^3 under the OV Hypothesis.*
 - *DIAMETER in intersection graphs of axis-parallel unit cubes in \mathbb{R}^3 under the OV Hypothesis.*
 - *DIAMETER in intersection graphs of axis-parallel line segments in \mathbb{R}^2 under the OV Hypothesis.*
 - *DIAMETER-2 in intersection graphs of axis-parallel hypercubes in \mathbb{R}^{12} under the Hyperclique Hypothesis.*

Our results imply lower bounds for approximations. (See Section 4.2 for a short proof.)

- **Corollary 2.** *Under the Orthogonal Vectors and Hyperclique Hypotheses, for all $\delta, \varepsilon > 0$ there is no $\mathcal{O}(n^{2-\delta})$ time $(4/3 - \varepsilon)$ -approximation for DIAMETER in intersection graphs of unit segments or congruent equilateral triangles in \mathbb{R}^2 , and no $(3/2 - \varepsilon)$ -approximation in intersection graphs of axis-parallel hypercubes in $\mathbb{R}^{\geq 12}$. Furthermore, for all $\delta > 0$ there is no $\mathcal{O}(n^{2-\delta} \text{poly}(1/\varepsilon))$ time approximation scheme that provides a $(1 + \varepsilon)$ -approximation for DIAMETER for any $\varepsilon > 0$ in intersection graphs of axis-parallel unit segments in \mathbb{R}^2 , or unit balls or axis-parallel unit cubes in \mathbb{R}^3 .*

Theorem 1 shows that sub-quadratic algorithms in many intersection graphs classes are unlikely to exist; one must wonder if such algorithms are possible at all? A notable case missing from our lower bounds are the case of unit disks; indeed, it is possible that unit disk graphs enjoy sub-quadratic diameter computation. More generally, it is an interesting open question whether intersection graphs of so-called pseudodisks admit sub-quadratic diameter algorithms. (Pseudodisks are objects bounded by Jordan curves such that the boundaries of any pair of objects have at most two intersection points.) We make a step towards resolving this problem with the following theorem for intersection graphs of axis-parallel unit squares – since axis-parallel unit squares are pseudodisks.

► **Theorem 3.** *There is an $\mathcal{O}(n \log n)$ algorithm for DIAMETER-2 in unit square graphs.*

The algorithm is based on the insight that the problem can be simplified to the following: given *skylines* A, B and a list of axis-parallel squares S , check whether each pair $(a, b) \in A \times B$ is covered by some square $s \in S$. Since any axis-parallel square $s \in S$ covers *intervals* in A and B , this problem in turn reduces to checking whether the union of $|S|$ rectangles covers the $A \times B$ grid. Using near-linear skyline computation [35], and a line sweep for the grid covering problem, we obtain a surprisingly simple $\mathcal{O}(n \log n)$ time algorithm (in contrast to the quadratic-time hardness in higher dimensions).

Organization. After some preliminaries and the introduction of the complexity-theoretic hypotheses used in the paper, we present our algorithm for unit squares in Section 3. Section 4 showcases our lower bound techniques. The lower bounds for unit segments, congruent equilateral triangles as well as for axis-parallel unit segments have a structure similar to two other lower bounds in Section 4, and they can be found in the full version.

2 Preliminaries

Let $G = (V, E)$ be a graph, and u and v be vertices in G . The distance from u to v is denoted by $\text{dist}_G(u, v)$, and equals the number of edges on the shortest path from u to v in G . The diameter of G is denoted by $\text{diam}(G)$ and equals to $\max_{u, v \in V} \text{dist}_G(u, v)$. The open and closed neighborhood of a vertex v are $N(v) = \{u \in V \mid uv \in E\}$ and $N[v] = \{v\} \cup N(v)$, respectively. Let $A, B \subseteq V$ be sets of vertices. The diameter of A and B is denoted by $\text{diam}_G(A, B) = \max_{(a, b) \in A \times B} \text{dist}_G(a, b)$. Finally, let $[n]$ denote the set $\{1, \dots, n\}$.

2.1 Hardness assumptions

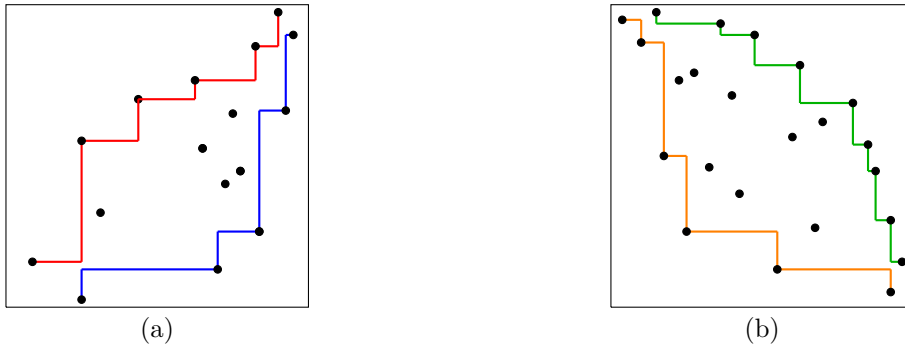
We use two hypotheses from fine-grained complexity theory for our lower bounds. For an overview of this field, we refer to the survey [44].

Orthogonal Vectors Hypothesis. Let OV denote the following problem: Given sets A, B of n vectors in $\{0, 1\}^d$, determine whether there exists an *orthogonal pair* $a \in A, b \in B$, i.e., for all $i \in [d]$ we have $(a)_i = 0$ or $(b)_i = 0$. Exhaustive search yields an $\mathcal{O}(n^2 d)$ algorithm, which can be improved for small dimension $d = c \log n$ to $\mathcal{O}(n^{2-1/O(\log(c))})$ [2, 18]. For larger dimensions $d = \omega(\log n)$, it is known [45] that no $\mathcal{O}(n^{2-\epsilon})$ -time algorithm can exist unless the Strong Exponential Time Hypothesis [32] fails. Thus, the Strong Exponential Time Hypothesis implies the following (so-called “moderate-dimensional”) OV Hypothesis.

► **Hypothesis 4 (Orthogonal Vectors Hypothesis).** *For no $\epsilon > 0$, there is an algorithm that solves OV in time $\mathcal{O}(\text{poly}(d)n^{2-\epsilon})$.*

By now, there is an extensive list of problems with tight lower bounds (including sub-quadratic equivalences) based on this assumption, see [44].

Hyperclique Hypothesis. For $k \geq 4$, let 3-UNIFORM k -HYPERCLIQUE denote the following problem: Given a 3-uniform hypergraph $G = (V, E)$, determine whether there exists a *hyperclique* of size k , i.e., a set $S \subseteq V$ such that for all $e \in \binom{S}{3}$, we have $e \in E$. By exhaustive search, we can solve this problem in time $\mathcal{O}(n^k)$ where $n = |V|$. Unlike the usual k -CLIQUE problem in graphs, for which a $\mathcal{O}(n^{\omega k/3 + \mathcal{O}(1)})$ algorithm exists [39], no techniques are known that would beat exhaustive search by a polynomial factor for the problem in hypergraphs. This has led to the hypothesis that exhaustive search is essentially best possible.



■ **Figure 1** The skylines (or fronts) of a point set P . In figure (a), the points on the blue curve are $\text{BRF}(P)$ and on the red curve are $\text{TLF}(P)$. In figure (b), the points on the green curve are $\text{TRF}(P)$ and on the orange curve are $\text{BLF}(P)$.

► **Hypothesis 5 (Hyperclique Hypothesis).** *For no $\epsilon > 0$ and $k \geq 4$, there is an algorithm that would solve 3-UNIFORM k -HYPERCLIQUE in time $\mathcal{O}(n^{k-\epsilon})$.*

See [38] for a detailed description of the plausibility of this hypothesis. Tight conditional lower bounds (including fine-grained equivalences) have been obtained, e.g., in [1, 11, 36, 4].

3 Solving the Diameter-2 problem on unit square graphs

In this section, we are going to present an algorithm with running time $\mathcal{O}(n \log n)$ for the DIAMETER-2 problem for unit square graphs. For each unit square $v \in V$, we consider the center of v , denoted \dot{v} , as the point representing v in the plane; for a square set $X \subset V$, we use \dot{X} to denote the set of corresponding centers. Let $\dot{G} = (\dot{V}, E)$ denote the graph on centers of squares in G . Hence, for all $\{u, v\} \in E(G)$, there is an edge between \dot{u} and \dot{v} . Note that we will often use \dot{G} and G interchangeably.

Notice that a graph has diameter at most two if and only if for every pair of vertices $u, v \in V$: $N[u] \cap N[v] \neq \emptyset$, i.e., there is a square w that both u and v have an intersection with or they intersect each other. Equivalently, the square of side length 2 centered at \dot{w} must cover both \dot{u} and \dot{v} . For a square w , let w^2 denote the side-length-2 square of center \dot{w} . Thus, in order to decide whether $\text{diam}(G) \leq 2$, it is sufficient to check whether for every $u, v \in V$ there exists $w \in V$ such that $\dot{u}, \dot{v} \in w^2$.

For a set of points P we define the *top-left front*, $\text{TLF}(P)$, and *bottom-right front*, $\text{BRF}(P)$ as follows (see Figure 1a).

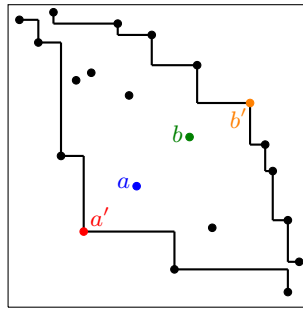
$$\begin{aligned} \text{TLF}(P) &= \{p \in P \mid \forall q \in P: p_x \leq q_x \text{ or } p_y \geq q_y\} \\ \text{BRF}(P) &= \{p \in P \mid \forall q \in P: p_x \geq q_x \text{ or } p_y \leq q_y\} \end{aligned}$$

Similarly, we define the *top-right front*, $\text{TRF}(P)$, and *bottom-left front*, $\text{BLF}(P)$ as follows (see Figure 1b).

$$\begin{aligned} \text{TRF}(P) &= \{p \in P \mid \forall q \in P: p_x \geq q_x \text{ or } p_y \geq q_y\} \\ \text{BLF}(P) &= \{p \in P \mid \forall q \in P: p_x \leq q_x \text{ or } p_y \leq q_y\} \end{aligned}$$

► **Lemma 6.** *The graph G has diameter at most 2 if and only if*

$$\max \left(\text{diam}_{\dot{G}}(\text{BLF}(\dot{V}), \text{TRF}(\dot{V})), \text{diam}_{\dot{G}}(\text{TLF}(\dot{V}), \text{BRF}(\dot{V})) \right) \leq 2.$$



■ **Figure 2** Any square covering a' and b' also covers a and b .

Proof. If G has diameter at most two, then clearly any pair of subsets of \dot{V} have diameter at most 2 in \dot{G} . For the other direction, consider any pair $a, b \in \dot{V}$, and assume that $a_x \leq b_x$ and $a_y \leq b_y$. We prove that $\text{dist}_{\dot{G}}(a, b) \leq 2$.

Select $a' \in \text{BLF}(\dot{V})$ such that $a'_x \leq a_x$ and $a'_y \leq a_y$, see Figure 2. Similarly, select $b' \in \text{TRF}(\dot{V})$ such that $b_x \leq b'_x$ and $b_y \leq b'_y$. Then we can observe that the minimum bounding box of $\{a', b'\}$ covers the minimum bounding box of $\{a, b\}$. Since $\text{dist}_{\dot{G}}(a', b') \leq \text{diam}_{\dot{G}}(\text{BLF}(\dot{V}), \text{TRF}(\dot{V})) \leq 2$, there exists a square $w \in V$ such that w^2 covers $\{a', b'\}$. Consequently, w^2 also covers $\{a, b\}$, and thus $\text{dist}_{\dot{G}}(a, b) \leq 2$.

Finally, the case $a_x > b_x$ and $a_y > b_y$ is symmetric, and the cases $a_x > b_x, a_y \leq b_y$ and $a_x \leq b_x, a_y > b_y$ are analogous with TLF and BRF instead of TRF and BLF. ◀

Using Lemma 6, we are able to prove Theorem 3.

Proof of Theorem 3. We start our algorithm by computing $\text{TLF}(\dot{V}), \text{TRF}(\dot{V}), \text{BLF}(\dot{V})$, and $\text{BRF}(\dot{V})$ in $\mathcal{O}(n \log n)$ time [35]. Let $\dot{P} = \text{BLF}(\dot{V})$ and $\dot{Q} = \text{TRF}(\dot{V})$. By Lemma 6, it is sufficient to show that in $\mathcal{O}(n \log n)$ time we can decide whether $\text{diam}_{\dot{G}}(\dot{P}, \dot{Q}) \leq 2$; using the same algorithm for $\text{BRF}(\dot{V})$ and $\text{TLF}(\dot{V})$ will then get the desired running time.

In order to check whether $N[\dot{p}] \cap N[\dot{q}] \neq \emptyset$ for all $(\dot{p}, \dot{q}) \in \dot{P} \times \dot{Q}$, we do the following: Consider $\dot{P} = \{\dot{p}_1, \dots, \dot{p}_{|\dot{P}|}\}$ and $\dot{Q} = \{\dot{q}_1, \dots, \dot{q}_{|\dot{Q}|}\}$ in x -order. Also, let $\text{GRID} = [|\dot{P}|] \times [|\dot{Q}|]$ be a grid where \dot{p}_i corresponds to the i -th row and \dot{q}_j corresponds to the j -th column.

For each square $v \in V$, recall that v^2 denotes the square with the same center but twice the side length. For each $v \in V$, define $I_v \subseteq \{1, 2, \dots, |\dot{P}|\}$ such that $i \in I_v$ iff v^2 contains \dot{p}_i . Similarly, $J_v \subseteq \{1, 2, \dots, |\dot{Q}|\}$ such that $j \in J_v$ iff v^2 contains \dot{q}_j . Since v^2 is an axis-parallel square, it covers intervals from both \dot{P} and \dot{Q} , thus I_v and J_v consist of consecutive integers. Therefore, we can think of the sets $I_v \times J_v$ as rectangles in GRID .

▷ **Claim 7.** We have $N[\dot{p}] \cap N[\dot{q}] \neq \emptyset$ for all $(\dot{p}, \dot{q}) \in \dot{P} \times \dot{Q}$ if and only if the union of $I_v \times J_v$ over all squares $v \in V$ covers GRID .

Proof. If the union of all rectangles covers the whole grid, then for any pair $(\dot{p}_i, \dot{q}_j) \in \dot{P} \times \dot{Q}$ of centers, there is a rectangle $I_v \times J_v$ that covers (i, j) . Therefore, v^2 covers both \dot{p}_i and \dot{q}_j . Thus, v is a shared neighbor of \dot{p} and \dot{q} .

If $N[\dot{p}] \cap N[\dot{q}] \neq \emptyset$ for all $(\dot{p}, \dot{q}) \in \dot{P} \times \dot{Q}$, then for each pair (\dot{p}_i, \dot{q}_j) there is at least one square v_{ij} such that v_{ij}^2 contains both \dot{p}_i and \dot{q}_j . Hence, $(i, j) \in I_{v_{ij}} \times J_{v_{ij}}$ for each $(i, j) \in \text{GRID}$. As a result, the union of $I_v \times J_v$ over all squares v^2 covers GRID . ◀

Note that the problem in Claim 7 corresponds to determining whether a union of rectangles covers the full grid. This problem can be solved in $\mathcal{O}(n \log n)$ time with a plane sweep [6, 43]. The time needed to construct the rectangles in GRID is $\mathcal{O}(n \log n)$ as there are $\mathcal{O}(n)$ rectangles. This concludes the proof of Theorem 3. ◀

4 Lower bounds based on the Orthogonal Vectors Hypothesis

In this section, we prove lower bounds for finding the diameter in various intersection graphs.

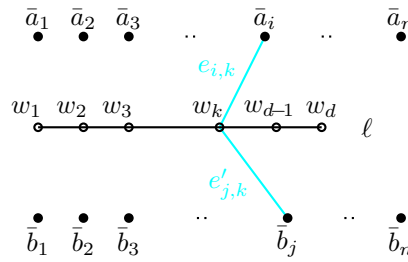
For a comparison to similar results on sparse graphs, let us briefly describe the result ruling out a $(3/2 - \epsilon)$ -approximation in time $\mathcal{O}(n^{2-\delta})$, for any $\epsilon, \delta > 0$, due to Roditty and Vassilevska-Williams [40]. While it is originally stated as a reduction from k -Dominating Set, we adapt it to give a reduction from OV: Given sets $A, B \subseteq \{0, 1\}^d$, introduce *vector nodes* for each $a \in A$ and $b \in B$ as well as *coordinate nodes* for $k \in [d]$. Without loss of generality (see Section 4.1), one may assume that all vectors $a \in A$ have $(a)_{d-1} = 1$ and all vectors $b \in B$ have $(b)_d = 1$. We connect each vector node $v \in A \cup B$ to the coordinate node $k \in [d]$ iff $(v)_k = 1$, and make all coordinate nodes a clique by adding all possible edges between coordinate nodes. The important observation is that (1) a pair $a \in A, b \in B$ has distance at most 2 iff there is a $k \in [d]$ such that $(a)_k = (b)_k = 1$, i.e., a, b do *not* form an orthogonal pair, and (2) all other types of node pairs have distance at most 2. Thus, A, B contains an orthogonal pair iff the diameter of the constructed graph is at least 3. Since the reduction produces a sparse graph with $\mathcal{O}(n + d)$ nodes and $\mathcal{O}(nd)$ edges in time $\mathcal{O}(nd)$, any $\mathcal{O}(m^{2-\delta})$ -time algorithm distinguishing between diameter 2 and 3 would give a $\mathcal{O}(n^{2-\delta} \text{poly}(d))$ -time OV algorithm, refuting the OV Hypothesis.

Generally speaking, implementing this reduction using low-dimensional geometric graphs is problematic: we must be able to implement an arbitrary bipartite graph on a vertex set $L \times R$ where $|L| = n$ and $|R| = d$. Instead, in this section we implement two different types of reductions via geometric graphs; the main ideas are as follows:

Diameter-3 graphs (Section 4.1 and full version). Instead of *coordinate nodes*, we introduce *1-entry nodes* $(v)_k$ for all $v \in A \cup B, k \in [d]$ with $(v)_k = 1$. This increases the number of nodes only to $\mathcal{O}(nd)$, while allowing us to geometrically implement edges of the form $\{v, (v)_k\}$ for all $v \in A \cup B, k \in [d]$ with $(v)_k = 1$ and $\{(v)_k, (v')_k\}$ for all $v, v' \in A \cup B, k \in [d]$ with $(v)_k = (v')_k = 1$. Now, a witness of non-orthogonality of a, b is a 3-path $a - (a)_k - (b)_k - b$. By showing that all other distances are bounded by 3, we obtain hardness for the DIAMETER-3 problem. See Section 4.1 and the full version for details, including the use of an additional node to make all 1-entry nodes sufficiently close in distance.

(Non-sparse) Diameter- $\Theta(d)$ graphs (Section 4.2 and full version). Instead of *coordinate nodes* or *1-entry nodes*, we introduce *vector-coordinate nodes* $(v)_k$ for all $v \in A \cup B, k \in [d]$, irrespective of whether $(v)_k = 1$. As opposed to previously, we do not create a constant diameter instance: The idea is to create an instance where the most distant pairs are of the form $(a)_1, (b)_d$ for $a \in A, b \in B$, and a non-orthogonality witness is a path of the form $(a)_1 \rightsquigarrow \dots \rightsquigarrow (a)_k \rightsquigarrow (b)_k \rightsquigarrow \dots \rightsquigarrow (b)_d$ with $(a)_k = (b)_k = 1$. This construction requires us to implement perfect matchings between vector-coordinate gadgets $(a)_k$ for $a \in A$ and $(a')_{k+1}$ for $a' \in A$ if $a = a'$, as well as a gadget for implementing short connections for $(a)_k \rightsquigarrow (b)_k$ that check whether $(a)_k = (b)_k = 1$. Interestingly, this type of reduction generally produces dense graphs with $\Omega(n^2)$ edges, so this approach crucially exploits the expressive power of geometric graphs to give a subquadratic reduction. See Section 4.2 and the full version for details, including a description of auxiliary nodes not mentioned here.

Finally, we remark that the reduction for unit hypercubes given in Section 5 has the most similar structure to the reduction by Roditty and Vassilevska-Williams [40], despite starting from a different hypothesis, and has similarities to [4, Theorem 14]. We crucially exploit properties of the hyperclique problem to implement it using hypercube graphs.



■ **Figure 3** Reducing orthogonal vectors to DIAMETER-3 in intersection graphs of line segments.

4.1 The Diameter-3 problem for line segment intersection graphs

In this section, we are going to present a lower bound on the running time of the algorithm for the DIAMETER-3 problem for line segment intersection graphs, such that vertices are line segments with any length, and there is an edge between a pair of line segments if they intersect. This serves as a warm-up for the slightly more complicated reductions below.

► **Theorem 8.** *For all $\epsilon > 0$, there is no $\mathcal{O}(n^{2-\epsilon})$ time algorithm for the DIAMETER-3 problem for line segment intersection graphs, unless the OV Hypothesis fails.*

Let $A = \{a_1, a_2, \dots, a_n\}$ and $B = \{b_1, b_2, \dots, b_n\}$ be two sets of n vectors in $\{0, 1\}^d$. We construct a set of segments such that the diameter of the corresponding intersection graph is at most 3 if and only if there is no orthogonal pair $(a, b) \in A \times B$.

Without loss of generality, we assume that for each $a_i \in A$ and $b_j \in B$, $((a_i)_{d-1}, (a_i)_d) = (1, 0)$ and $((b_j)_{d-1}, (b_j)_d) = (0, 1)$, by adding two coordinates to the ends of the vectors. Note that adding these coordinates does not change whether vectors a, b are orthogonal or not.

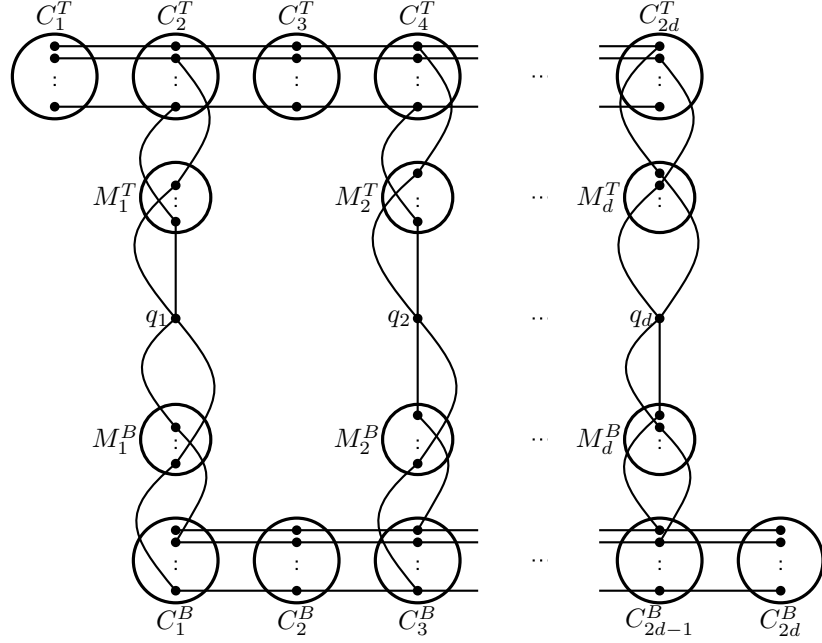
For each vector $a_i \in A$, let \bar{a}_i denote a zero-length line segment from $(i, 1)$ to $(i, 1)$. Analogously, for each vector $b_j \in B$, let \bar{b}_j denote a line segment from $(j, -1)$ to $(j, -1)$. Furthermore, let ℓ be a line segment from $(1, 0)$ to $(d, 0)$, and let $\{w_1, w_2, \dots, w_d\}$ be d different points on ℓ such that for all $k \in [d]$, w_k is located at $(k, 0)$. Moreover, for each $a_i \in A$, if $(a_i)_k = 1$, we define a line segment $e_{i,k}$ from \bar{a}_i to w_k (i.e., from $(i, 1)$ to $(k, 0)$). Analogously, for each $b_j \in B$, if $(b_j)_{k'} = 1$, we define a line segment $e'_{j,k'}$ from \bar{b}_j to $w_{k'}$ (i.e., from $(j, -1)$ to $(k', 0)$). Let \bar{V} be the set of constructed line segments, and let G be their intersection graph (see Figure 3).

► **Lemma 9.** *The sets A and B contain an orthogonal pair if and only if $\text{diam}(G) \geq 4$.*

Let \bar{A} be the set of line segments corresponding to vectors in A . Analogously, let \bar{B} be the set of line segments corresponding to vectors in B . To prove the lemma, we show that each pair of vertices is within distance at most 3, unless it is in $\bar{A} \times \bar{B}$ (see Claim 10 below and see the full version for its proof). The pairs in $\bar{A} \times \bar{B}$ have distance 4 or 3 depending on whether their corresponding vectors in $A \times B$ are orthogonal or not.

▷ **Claim 10.** $\text{dist}(\bar{u}, \bar{v}) \leq 3$ for all $(\bar{u}, \bar{v}) \in (\bar{V} \times \bar{V}) \setminus (\bar{A} \times \bar{B} \cup \bar{B} \times \bar{A})$.

Proof of Lemma 9. If a_i and b_j are not orthogonal, then there is at least one $k \in [d]$ such that $(a_i)_k = (b_j)_k = 1$. Hence, the path $\bar{a}_i - e_{i,k} - e'_{j,k} - \bar{b}_j$ exists, and it has length 3. If a_i and b_j are orthogonal, then there is no index k such that $(a_i)_k = (b_j)_k = 1$. Consequently, there is no path of length 3 from \bar{a}_i to \bar{b}_j , and $\text{dist}(\bar{a}_i, \bar{b}_j) \geq 4$. Together with Claim 10 this proves the lemma. ◀



■ **Figure 4** Schematic picture of the graph $G(A)$.

Proof of Theorem 8. The above reduction creates a set of $N = \mathcal{O}(nd)$ segments in $\mathcal{O}(nd)$ time. If there is an algorithm solving DIAMETER-3 in $\mathcal{O}(N^{2-\delta})$ time in segment intersection graphs, then combining this algorithm with the reduction would solve OV in time $\mathcal{O}(nd) + \mathcal{O}((nd)^{2-\delta}) = \mathcal{O}(n^{2-\delta}\text{poly}(d))$, refuting the OV Hypothesis. ◀

4.2 The diameter problem for unit ball graphs

► **Theorem 11.** *For all $\epsilon > 0$, there is no $\mathcal{O}(n^{2-\epsilon})$ time algorithm for solving DIAMETER in unit ball graphs in \mathbb{R}^3 under the Orthogonal Vectors Hypothesis.*

Let $A = \{a_1, a_2, \dots, a_n\}$ be a given set of vectors from $\{0, 1\}^d$. First, we construct graph $G(A)$ and show that $G(A)$ has diameter $\geq 2d + 5$ if and only if there is an orthogonal pair of vectors in A . Next, we show how $G(A)$ can be realized as an intersection graph of unit balls in \mathbb{R}^3 . Without loss of generality, assume that the all-one vector is an element of A (if it is not in A , then adding the all-one vector does not change whether there is an orthogonal pair.)

We construct a graph $G(A)$ as follows. Let C_1^T, \dots, C_{2d}^T and C_1^B, \dots, C_{2d}^B be cliques, such that for all $k \in [2d]$, $C_k^T = \{v_{k,1}^T, \dots, v_{k,n}^T\}$, $C_k^B = \{v_{k,1}^B, \dots, v_{k,n}^B\}$, and $v_{k,i}^T$ and $v_{k,i}^B$ correspond to a_i for all $i \in [n]$, see Figure 4. We add a perfect matching between each pair C_k^T and C_{k+1}^T for all $k \in [2d-1]$ such that there is an edge incident to $v_{k,i}^T$ and $v_{k+1,i}^T$ for all $i \in [n]$. Analogously, there is a perfect matching between each pair C_k^B and C_{k+1}^B .

Let M_1^T, \dots, M_d^T be cliques such that if $(a_i)_k = 1$, then there is a vertex $m_{k,i}^T$ in M_k^T that is adjacent to $v_{k,i}^T$. Similarly, let M_1^B, \dots, M_d^B be cliques such that if $(a_i)_k = 1$, then there is a vertex $m_{k,i}^B$ in M_k^B that is adjacent to $v_{k,i}^B$. Notice that because of the addition of the all ones vector, the cliques M_k^T and M_k^B are all non-empty.

Finally, let $Q = \{q_1, q_2, \dots, q_d\}$ be a set of vertices such that q_k has edges to all vertices in M_k^T and M_k^B for all $k \in [d]$.

► **Lemma 12.** *The graph $G(A)$ has diameter at most $2d + 4$ iff A has no orthogonal pair.*

Proof. Assume that there is an orthogonal pair $(a_i, a_j) \in A$ such that $i \neq j$. Hence, $\sum_{k=1}^n (a_i)_k (a_j)_k = 0$, which means that there is no $k \in [n]$ such that $(a_i)_k = (a_j)_k = 1$. Consequently, for all $k \in [d]$, the distance from $v_{2k,i}^T \in C_{2k}^T$ to $v_{2k-1,j}^B \in C_{2k-1}^B$ is at least 5. Therefore, $2d + 4 < \text{dist}(v_{1,i}^T, v_{2d,j}^B) \leq \text{diam}(G(A))$.

Now suppose that A has no orthogonal pair. We want to prove that $\text{diam}(G(A)) \leq 2d + 4$. Since A has no orthogonal pair, for each pair (a_i, a_j) there is at least one $k \in [n]$ such that $(a_i)_k = (a_j)_k = 1$. Therefore, there are cliques M_k^T and M_k^B that have the vertices $m_{k,i}^T$ and $m_{k,j}^B$, respectively. Since all vertices in M_k^T and M_k^B have an edge to q_k , we can reach q_k from $v_{1,i}^T$ by a path of length $2k - 1 + 2$. Simultaneously, we can reach q_k from $v_{2d,j}^B$ by a path of length $2d - 2k + 1 + 2$. In total, this gives a path of length $2d + 4$ between $v_{1,i}^T$ and $v_{2d,j}^B$. Furthermore, it is easy check that the distance of any pair of vertices where at least one vertex is outside $C_1^T \cup C_{2d}^B$ is at most $2d + 4$. As a result, $\text{diam}(G(A)) \leq 2d + 4$. ◀

► **Lemma 13.** $G(A)$ can be realized as an intersection graph of unit balls in \mathbb{R}^3 .

Proof. For converting $G(A)$ into an intersection graph of unit balls, we should consider each vertex in $G(A)$ as the center of a unit diameter ball, and for those vertices that are adjacent, their corresponding unit balls should intersect. To this end, we choose the following coordinates for the centers of the unit balls in \mathbb{R}^3 :

- For all $k \in [2d]$ and $i \in [n]$, the center point of $v_{k,i}^T \in C_k^T$ is $(k, \frac{i}{n}, 0)$.
- For all $k \in [d]$ and $i \in [n]$, if $m_{k,i}^T \in M_k^T$ exists, then its center point is $(2k, \frac{i}{n}, -1)$.
- For all $k \in [d]$, the center point of $q_k \in Q$ is $(2k, \frac{1}{2}, -1.6)$.
- For all $k \in [d]$ and $i \in [n]$, if $m_{i,j}^B \in M_i^B$ exists, its center point is $(2k, \frac{i}{n}, -2.2)$.
- For all $k \in [2d]$ and $i \in [n]$, the center point of $v_{k,i}^B \in C_k^B$ is $(k, \frac{i}{n}, -3.2)$.

The distance between center points that correspond to adjacent vertices should be at most 1. For each two vertices in the same clique in C_k^T , C_k^B , M_k^T , and M_k^B their center points differ only in the y -coordinate. Since this difference is at most $1 - 1/n < 1$, they form a clique. For each two adjacent vertices in two different cliques, their center points differ either only in the x -, or only in the z -coordinate, by exactly 1, hence, they intersect. For a vertex in Q and M_k^T , if $m_{k,i}^T$ exists, the distance between $m_{k,i}^T$ and q_k is

$$\sqrt{(2k - 2k)^2 + (\frac{i}{n} - \frac{1}{2})^2 + (-1 - (-1.6))^2} = \sqrt{(\frac{i}{n} - \frac{1}{2})^2 + (0.6)^2} \leq \sqrt{(\frac{1}{2})^2 + (0.6)^2} < 1$$

The same argument holds for adjacent vertices in Q and M^B . One can easily check that the non-adjacent vertices have distance strictly greater than 1. ◀

Proof of Theorem 11. The construction creates a set of $N = \mathcal{O}(nd)$ balls in $\mathcal{O}(nd)$ time. If there is an algorithm to solve DIAMETER in $\mathcal{O}(N^{2-\delta})$ time in ball graphs, then we could combine this construction with the algorithm, and solve the ORTHOGONAL VECTORS problem in $\mathcal{O}(nd) + \mathcal{O}((nd)^{2-\delta}) = \mathcal{O}(n^{2-\delta} \text{poly}(d))$ time. This contradicts the Orthogonal Vectors Hypothesis, and concludes the theorem. ◀

A simple transformation of this construction shows that we can realize $G(A)$ also as an intersection graph of axis-parallel unit cubes.

► **Corollary 14.** For all $\epsilon > 0$, there is no $\mathcal{O}(n^{2-\epsilon})$ time algorithm for solving DIAMETER in intersection graphs of axis-parallel unit cubes in \mathbb{R}^3 under the Orthogonal Vectors Hypothesis.

Proof. Let P denote the set of centers constructed for unit balls. We rotate P by $\pi/4$ around the y axis, and scale P by a factor of $\sqrt{2}$. Let P' be the resulting set of points. Note that in P , all inter-clique edges were realized by a horizontal or vertical point pair of distance

exactly 1. In P' , the corresponding pairs are diagonal segments in some plane perpendicular to the y -axis, therefore the unit side-length cubes centered at the corresponding pair of points will have a touching edge. It is routine to check that the unit side-length cubes centered at P' realize the intersection graph $G(A)$. ◀

Proof of Corollary 2. The lower bounds regarding constant-approximations in sub-quadratic time are immediate consequences of our lower bounds for DIAMETER-2 and DIAMETER-3. Notice that our proofs for unit balls and axis-parallel unit cubes in \mathbb{R}^3 , as well as axis-parallel unit segments in \mathbb{R}^2 use a construction where the resulting intersection graph has diameter $d^* = \Theta(d)$. Under OV, there exists no $(1 + \varepsilon)$ -approximation for these problems that would run in $n^{2-\delta} \text{poly}(1/\varepsilon)$ time, as setting $\varepsilon = 1/d^* = \Theta(1/d)$ would enable us to decide OV in $n^{2-\delta} \text{poly}(d)$ time. ◀

5 The Diameter-2 problem for hypercube graphs: a hyperclique lower bound

► **Theorem 15.** *For all $\epsilon > 0$ there is no $O(n^{2-\epsilon})$ algorithm for DIAMETER-2 in unit hypercube graphs in \mathbb{R}^{12} , unless the Hyperclique Hypothesis fails.*

Proof. Observe that under the Hyperclique Hypothesis, it requires time $n^{6-o(1)}$ to find a hyperclique of size 6 in a given 3-uniform hypergraph $G = (V, E)$. In fact, using a standard color-coding argument, we can assume without loss of generality that G is 6-partite: We have $V = V_1 \cup \dots \cup V_6$ for disjoint sets V_i of size n each, and any 6-hyperclique must choose exactly one vertex from each V_i . By slight abuse of notation, we view each V_i as a disjoint copy of $[n]$, i.e., node $j \in [n]$ in V_i is different from node j in $V_{i'}$ with $i' \neq i$. Furthermore, by complementing the edge set, we arrive at the equivalent task of determining whether G has an *independent set* of size 6, i.e., whether there are $(v_1, \dots, v_6) \in V_1 \times \dots \times V_6$ such that $\{v_i, v_j, v_k\} \notin E$ for all distinct $i, j, k \in [6]$. Finally, for technical reasons, we assume without loss of generality that for each $v_i \in V_i$ and distinct $j, k \in [6] \setminus \{i\}$, there are $v_j \in V_j, v_k \in V_k$ with $\{v_i, v_j, v_k\} \in E$: To this end, simply add, for every $\ell \in [6]$, a dummy vertex v'_ℓ to V_ℓ , and add, for every i, j, k and $v_j \in V_j, v_k \in V_k$, the edge $\{v'_i, v_j, v_k\}$ to E , i.e., each dummy vertex is connected to all other pairs of vertices (including other dummy vertices). Observe that this yields an equivalent instance, since no dummy vertex can be contained in an independent set.

The reduction is given by constructing a set of $O(n^3)$ unit hypercubes in \mathbb{R}^{12} , which we specify by their centers. These (hyper)cubes are of three types: *left-half cubes* representing a choice of the vertices $(x_1, x_2, x_3) \in V_1 \times V_2 \times V_3$, *right-half cubes* representing a choice of the vertices $(y_1, y_2, y_3) \in V_4 \times V_5 \times V_6$ and *edge cubes* representing an edge $\{v_i, v_j, v_k\} \in E$. In particular, the choice of a vertex in V_i will be encoded in the dimensions $2i - 1$ and $2i$.

Specifically, for each $(x_1, x_2, x_3) \in V_1 \times V_2 \times V_3$ such that $\{x_1, x_2, x_3\} \notin E$, we define the center of the left-half cube X_{x_1, x_2, x_3} as

$$\left(\frac{x_1}{n+1}, 1 - \frac{x_1}{n+1}, \frac{x_2}{n+1}, 1 - \frac{x_2}{n+1}, \frac{x_3}{n+1}, 1 - \frac{x_3}{n+1}, 2, \dots, 2 \right).$$

Similarly, for each $(y_1, y_2, y_3) \in V_4 \times V_5 \times V_6$ such that $\{y_1, y_2, y_3\} \notin E$, we define the center of the right-half cube Y_{y_1, y_2, y_3} as

$$\left(2, \dots, 2, \frac{y_1}{n+1}, 1 - \frac{y_1}{n+1}, \frac{y_2}{n+1}, 1 - \frac{y_2}{n+1}, \frac{y_3}{n+1}, 1 - \frac{y_3}{n+1} \right).$$

Finally, for each edge $e = \{v_i, v_j, v_k\} \in E$ not already in $V_1 \times V_2 \times V_3 \cup V_4 \times V_5 \times V_6$, we define a corresponding edge cube E_{v_i, v_j, v_k} with the following center point: We set the $2i - 1$ -th coordinate to $1 + \frac{v_i}{n+1}$, the $2i$ -th coordinate to $2 - \frac{v_i}{n+1}$, and similarly we set the coordinates $2j - 1, 2j, 2k - 1, 2k$ to $1 + \frac{v_j}{n+1}, 2 - \frac{v_j}{n+1}, 1 + \frac{v_k}{n+1}, 2 - \frac{v_k}{n+1}$, respectively, and we set all remaining coordinates to 1. For example, if $i = 1, j = 2, k = 4$, the center point of E_{v_1, v_2, v_4} is

$$\left(1 + \frac{v_1}{n+1}, 2 - \frac{v_1}{n+1}, 1 + \frac{v_2}{n+1}, 2 - \frac{v_2}{n+1}, 1, 1, 1 + \frac{v_4}{n+1}, 2 - \frac{v_4}{n+1}, 1, 1, 1, 1\right).$$

Let S denote the set of all unit cubes $X_{x_1, x_2, x_3}, Y_{y_1, y_2, y_3}, E_{v_i, v_j, v_k}$ constructed above and let G_S denote the geometric intersection graph of the unit cubes. We prove that $\text{diam}(G_S) \leq 2$ if and only if there is no independent set $(v_1, \dots, v_6) \in V_1 \times \dots \times V_6$ in the 3-uniform hypergraph $G = (V_1 \cup \dots \cup V_6, E)$:

1. **Intra-set distances:** We have that the left- and right-half cubes as well as the edge cubes form cliques, i.e., $\text{dist}_{G_S}(X_{x_1, x_2, x_3}, X_{x'_1, x'_2, x'_3}) \leq 1$, $\text{dist}_{G_S}(Y_{y_1, y_2, y_3}, Y_{y'_1, y'_2, y'_3}) \leq 1$ and $\text{dist}_{G_S}(E_{v_1, v_2, v_3}, E_{v'_1, v'_2, v'_3}) \leq 1$: Observe that the center of each X_{x_1, x_2, x_3} is contained in $[0, 1]^6 \times \{2\}^6$ and thus in a hypercube of side length at most 1. Thus, all cubes X_{x_1, x_2, x_3} intersect each other, proving $\text{dist}_{G_S}(X_{x_1, x_2, x_3}, X_{x'_1, x'_2, x'_3}) \leq 1$. The remaining claims follow analogously by observing that the centers of Y_{y_1, y_2, y_3} and E_{v_1, v_2, v_3} are contained in $\{2\}^6 \times [0, 1]^6$ and $[1, 2]^{12}$, respectively, and thus also in hypercubes of side length at most 1.
2. **Equality checks:** Let $x_1 \in V_1, x_2 \in V_2, x_3 \in V_3$ and $v_i \in V_i, v_j \in V_j, v_k \in V_k$. Then $\text{dist}_{G_S}(X_{x_1, x_2, x_3}, E_{v_i, v_j, v_k}) = 1$ iff $v_\ell = x_\ell$ whenever $\ell \in \{1, 2, 3\} \cap \{i, j, k\}$: Consider $\ell \in \{1, 2, 3\} \cap \{i, j, k\}$. Then the dimensions $(2\ell - 1, 2\ell)$ of X_{x_1, x_2, x_3} and E_{v_i, v_j, v_k} are equal to $(\frac{x_\ell}{n+1}, 1 - \frac{x_\ell}{n+1})$ and $(1 + \frac{v_\ell}{n+1}, 2 - \frac{v_\ell}{n+1})$, respectively. Note that $(1 + \frac{v_\ell}{n+1}) - \frac{x_\ell}{n+1} \leq 1$ and $(2 - \frac{v_\ell}{n+1}) - (1 - \frac{x_\ell}{n+1}) \leq 1$ hold simultaneously iff $x_\ell = v_\ell$. All other dimensions $\ell' \notin \{1, 2, 3\} \cap \{i, j, k\}$ are trivially within distance 1, since dimensions $(2\ell' - 1, 2\ell')$ of X_{x_1, x_2, x_3} and E_{v_i, v_j, v_k} are $(2, 2)$ and in $[1, 2]^2$, respectively (if $\ell' \notin \{1, 2, 3\}$), or in $[0, 2]^2$ and $(1, 1)$, respectively (if $\ell' \notin \{i, j, k\}$). The analogous claim holds for distances between Y_{y_1, y_2, y_3} and E_{v_i, v_j, v_k} .
3. **Edge distances:** We have that $\text{dist}_{G_S}(X_{x_1, x_2, x_3}, E_{v_i, v_j, v_k}) \leq 2$: By our technical assumption, we have that there is an edge $\{x_1, v'_4, v'_5\} \in E$ for some vertices $v'_4 \in V_4$ and $v'_5 \in V_5$. Thus, by the previous properties, we obtain that

$$\text{dist}_{G_S}(X_{x_1, x_2, x_3}, E_{v_i, v_j, v_k}) \leq \text{dist}_{G_S}(X_{x_1, x_2, x_3}, E_{x_1, v'_4, v'_5}) + \text{dist}_{G_S}(E_{x_1, v'_4, v'_5}, E_{v_i, v_j, v_k}) \leq 2.$$

4. **Distances of left- and right-half cubes:** Let $x_1 \in V_1, x_2 \in V_2, x_3 \in V_3$ and $y_1 \in V_4, y_2 \in V_5, y_3 \in V_6$ such that $\{x_1, x_2, x_3\}, \{y_1, y_2, y_3\} \notin E$ (thus, the left-half/right-half cubes for $\{x_1, x_2, x_3\}, \{y_1, y_2, y_3\}$ exist). Then we have that $\text{dist}_{G_S}(X_{x_1, x_2, x_3}, Y_{y_1, y_2, y_3}) > 2$ iff $(x_1, x_2, x_3, y_1, y_2, y_3)$ is an independent set in G : If the tuple $(x_1, x_2, x_3, y_1, y_2, y_3)$ is not an independent set, then there must be an edge $\{x_i, y_j, y_k\}$ or $\{x_i, x_j, y_k\}$ with $i, j, k \in [3]$, since $\{x_1, x_2, x_3\}$ and $\{y_1, y_2, y_3\}$ are non-edges. Consider the first case, the other is symmetric. Then by the equality-check property, that $\text{dist}_{G_S}(X_{x_1, x_2, x_3}, E_{x_i, x_j, y_k}) = 1$ and $\text{dist}_{G_S}(E_{x_i, x_j, y_k}, Y_{y_1, y_2, y_3}) = 1$, which yields $\text{dist}_{G_S}(X_{x_1, x_2, x_3}, Y_{y_1, y_2, y_3}) \leq 2$. It remains to consider the case that the tuple $(x_1, x_2, x_3, y_1, y_2, y_3)$ is an independent set. Since there cannot be any edge between a left-half cube $X_{x'_1, x'_2, x'_3}$ – which is contained in $(0, 1)^6 \times \{2\}^6$ – and a right-half cube $Y_{y'_1, y'_2, y'_3}$ – which is contained in $\{2\}^6 \times (0, 1)^6$ –, the only way to reach Y_{y_1, y_2, y_3} from X_{x_1, x_2, x_3} via a path of length 2 would have to use some edge cube E_{v_i, v_j, v_k} . However, by the equality-check property, a path $X_{x_1, x_2, x_3} - E_{v_i, v_j, v_k} - Y_{y_1, y_2, y_3}$ would

imply that the vertices chosen by $(x_1, x_2, x_3, y_1, y_2, y_3)$ would agree with v_i, v_j, v_k in the sets V_i, V_j, V_k . Thus, we would have found an edge $\{v_i, v_j, v_k\}$ among $(x_1, x_2, x_3, y_1, y_2, y_3)$, contradicting the assumption that it is an independent set.

Finally, observe that given a 3-uniform hypergraph G , we can construct the corresponding cube set S , containing $O(n^3)$ nodes, in time $O(n^3)$. Thus, if we had an $O(N^{2-\epsilon})$ -time algorithm for determining whether an N -vertex unit cube graph G_S has a diameter of at most 2, we could detect existence of an independent set (or equivalently, hyperclique) of size 6 in G in time $O(n^{6-3\epsilon})$, which would refute the Hyperclique Hypothesis. ◀

References

- 1 Amir Abboud, Karl Bringmann, Holger Dell, and Jesper Nederlof. More consequences of falsifying SETH and the orthogonal vectors conjecture. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018)*, pages 253–266. ACM, 2018. doi:10.1145/3188745.3188938.
- 2 Amir Abboud, Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '15*, pages 218–230. SIAM, 2015.
- 3 Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 28(4):1167–1181, 1999. doi:10.1137/S0097539796303421.
- 4 Haozhe An, Mohit Jayanti Gurumukhani, Russell Impagliazzo, Michael Jaber, Marvin Künnemann, and Maria Paula Parga Nina. The fine-grained complexity of multi-dimensional ordering properties. In Petr A. Golovach and Meirav Zehavi, editors, *Proc. 16th International Symposium on Parameterized and Exact Computation (IPEC 2021)*, volume 214 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.IPEC.2021.3.
- 5 Arturs Backurs, Liam Roditty, Gilad Segal, Virginia Vassilevska Williams, and Nicole Wein. Toward tight approximation bounds for graph diameter and eccentricities. *SIAM J. Comput.*, 50(4):1155–1199, 2021. doi:10.1137/18M1226737.
- 6 J. L. Bentley. Solutions to Klee’s rectangle problems. Unpublished manuscript, 1977.
- 7 Marthe Bonamy, Édouard Bonnet, Nicolas Bousquet, Pierre Charbit, Panos Giannopoulos, Eun Jung Kim, Pawel Rzazewski, Florian Sikora, and Stéphan Thomassé. EPTAS and subexponential algorithm for maximum clique on disk and unit ball graphs. *J. ACM*, 68(2):9:1–9:38, 2021. doi:10.1145/3433160.
- 8 Édouard Bonnet. 4 vs 7 sparse undirected unweighted diameter is SETH-hard at time $n^{4/3}$. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12–16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 34:1–34:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.34.
- 9 Édouard Bonnet. Inapproximability of diameter in super-linear time: Beyond the 5/3 ratio. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16–19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 17:1–17:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.17.
- 10 Andreas Brandstädt and Feodor F. Dragan. On linear and circular structure of (claw, net)-free graphs. *Discret. Appl. Math.*, 129(2-3):285–303, 2003. doi:10.1016/S0166-218X(02)00571-1.
- 11 Karl Bringmann, Nick Fischer, and Marvin Künnemann. A fine-grained analogue of Schaefer’s theorem in P: dichotomy of $\exists^k\forall$ -quantified first-order graph properties. In Amir Shpilka, editor, *Proc. 34th Computational Complexity Conference (CCC 2019)*, volume 137 of *LIPIcs*, pages 31:1–31:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CCC.2019.31.

- 12 Sergio Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. *ACM Trans. Algorithms*, 15(2):21:1–21:38, 2019. doi:10.1145/3218821.
- 13 Massimo Cairo, Roberto Grossi, and Romeo Rizzi. New bounds for approximating extremal distances in undirected graphs. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 363–376. SIAM, 2016. doi:10.1137/1.9781611974331.ch27.
- 14 Timothy M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46(2):178–189, 2003. doi:10.1016/S0196-6774(02)00294-8.
- 15 Timothy M. Chan and Dimitrios Skrepetos. All-pairs shortest paths in unit-disk graphs in slightly subquadratic time. In Seok-Hee Hong, editor, *27th International Symposium on Algorithms and Computation, ISAAC 2016, December 12-14, 2016, Sydney, Australia*, volume 64 of *LIPICs*, pages 24:1–24:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.ISAAC.2016.24.
- 16 Timothy M. Chan and Dimitrios Skrepetos. All-pairs shortest paths in geometric intersection graphs. *J. Comput. Geom.*, 10:27–41, 2019.
- 17 Timothy M. Chan and Dimitrios Skrepetos. Approximate shortest paths and distance oracles in weighted unit-disk graphs. *J. Comput. Geom.*, 10(2):3–20, 2019. doi:10.20382/jocg.v10i2a2.
- 18 Timothy M. Chan and Ryan Williams. Deterministic APSP, orthogonal vectors, and more: Quickly derandomizing Razborov-Smolensky. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '16*, pages 1246–1255. SIAM, 2016.
- 19 Shiri Chechik, Daniel H. Larkin, Liam Roditty, Grant Schoenebeck, Robert Endre Tarjan, and Virginia Vassilevska Williams. Better approximation algorithms for the graph diameter. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1041–1052. SIAM, 2014. doi:10.1137/1.9781611973402.78.
- 20 Derek G. Corneil. Lexicographic breadth first search - A survey. In Juraj Hromkovic, Manfred Nagl, and Bernhard Westfechtel, editors, *Graph-Theoretic Concepts in Computer Science, 30th International Workshop, WG 2004, Bad Honnef, Germany, June 21-23, 2004, Revised Papers*, volume 3353 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2004. doi:10.1007/978-3-540-30559-0_1.
- 21 Derek G. Corneil, Feodor F. Dragan, Michel Habib, and Christophe Paul. Diameter determination on restricted graph families. *Discret. Appl. Math.*, 113(2-3):143–166, 2001. doi:10.1016/S0166-218X(00)00281-X.
- 22 Derek G. Corneil, Feodor F. Dragan, and Ekkehard Köhler. On the power of BFS to determine a graph's diameter. *Networks*, 42(4):209–222, 2003. doi:10.1002/net.10098.
- 23 Mina Dalirrooyfard, Ray Li, and Virginia Vassilevska Williams. Hardness of approximate diameter: Now for undirected graphs. *CoRR*, abs/2106.06026, 2021. arXiv:2106.06026.
- 24 Mina Dalirrooyfard and Nicole Wein. Tight conditional lower bounds for approximating diameter in directed graphs. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1697–1710. ACM, 2021. doi:10.1145/3406325.3451130.
- 25 Mina Dalirrooyfard, Virginia Vassilevska Williams, Nikhil Vyas, and Nicole Wein. Tight approximation algorithms for bichromatic graph diameter and related problems. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 47:1–47:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ICALP.2019.47.
- 26 Feodor F. Dragan. Almost diameter of a house-hole-free graph in linear time via LexBFS. *Discret. Appl. Math.*, 95(1-3):223–239, 1999. doi:10.1016/S0166-218X(99)00077-3.
- 27 Feodor F. Dragan, Falk Nicolai, and Andreas Brandstädt. LexBFS-orderings and powers of graphs. In Fabrizio d'Amore, Paolo Giulio Franciosa, and Alberto Marchetti-Spaccamela, editors, *Graph-Theoretic Concepts in Computer Science*, pages 166–180. Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

- 28 Jie Gao and Li Zhang. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM J. Comput.*, 35(1):151–169, 2005. doi:10.1137/S0097539703436357.
- 29 Paweł Gawrychowski, Haim Kaplan, Shay Mozes, Micha Sharir, and Oren Weimann. Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{O}(n^{5/3})$ time. *SIAM J. Comput.*, 50(2):509–554, 2021. doi:10.1137/18M1193402.
- 30 Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985. doi:10.1145/2455.214106.
- 31 Harry B. Hunt III, Madhav V. Marathe, Venkatesh Radhakrishnan, S. S. Ravi, Daniel J. Rosenkrantz, and Richard Edwin Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms*, 26(2):238–274, 1998. doi:10.1006/jagm.1997.0903.
- 32 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.
- 33 Paul Koebe. *Kontaktprobleme der konformen Abbildung*. Hirzel, 1936.
- 34 Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Ad hoc networks beyond unit disk graphs. *Wirel. Networks*, 14(5):715–729, 2008. doi:10.1007/s11276-007-0045-6.
- 35 H. T. Kung, Fabrizio Luccio, and Franco P. Preparata. On finding the maxima of a set of vectors. *J. ACM*, 22(4):469–476, 1975. doi:10.1145/321906.321910.
- 36 Marvin Künnemann and Dániel Marx. Finding small satisfying assignments faster than brute force: A fine-grained perspective into boolean constraint satisfaction. In Shubhangi Saraf, editor, *Proc. 35th Computational Complexity Conference (CCC 2020)*, volume 169 of *LIPICs*, pages 27:1–27:28. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.CCC.2020.27.
- 37 Ray Li. Settling SETH vs. approximate sparse directed unweighted diameter (up to (NU)NSETH). In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21–25, 2021*, pages 1684–1696. ACM, 2021. doi:10.1145/3406325.3451045.
- 38 Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In Artur Czumaj, editor, *Proc. 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*, 2018. doi:10.1137/1.9781611975031.80.
- 39 Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, 026(2):415–419, 1985.
- 40 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1–4, 2013*, pages 515–524. ACM, 2013. doi:10.1145/2488608.2488673.
- 41 Raimund Seidel. On the all-pairs-shortest-path problem in unweighted undirected graphs. *J. Comput. Syst. Sci.*, 51(3):400–403, 1995. doi:10.1006/jcss.1995.1078.
- 42 Mikkel Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *J. ACM*, 46(3):362–394, 1999. doi:10.1145/316542.316548.
- 43 Jan van Leeuwen and Derick Wood. The measure problem for rectangular ranges in d-space. *J. Algorithms*, 2(3):282–300, 1981. doi:10.1016/0196-6774(81)90027-4.
- 44 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the International Congress of Mathematicians, ICM '18*, pages 3447–3487, 2018.
- 45 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005.

Computing Continuous Dynamic Time Warping of Time Series in Polynomial Time

Kevin Buchin  

Department of Computer Science, TU Dortmund, Germany

André Nusser  

BARC, University of Copenhagen, Denmark

Sampson Wong  

School of Computer Science, University of Sydney, Australia

Abstract

Dynamic Time Warping is arguably the most popular similarity measure for time series, where we define a time series to be a one-dimensional polygonal curve. The drawback of Dynamic Time Warping is that it is sensitive to the sampling rate of the time series. The Fréchet distance is an alternative that has gained popularity, however, its drawback is that it is sensitive to outliers.

Continuous Dynamic Time Warping (CDTW) is a recently proposed alternative that does not exhibit the aforementioned drawbacks. CDTW combines the continuous nature of the Fréchet distance with the summation of Dynamic Time Warping, resulting in a similarity measure that is robust to sampling rate and to outliers. In a recent experimental work of Brankovic et al., it was demonstrated that clustering under CDTW avoids the unwanted artifacts that appear when clustering under Dynamic Time Warping and under the Fréchet distance. Despite its advantages, the major shortcoming of CDTW is that there is no exact algorithm for computing CDTW, in polynomial time or otherwise.

In this work, we present the first exact algorithm for computing CDTW of one-dimensional curves. Our algorithm runs in time $\mathcal{O}(n^5)$ for a pair of one-dimensional curves, each with complexity at most n . In our algorithm, we propagate continuous functions in the dynamic program for CDTW, where the main difficulty lies in bounding the complexity of the functions. We believe that our result is an important first step towards CDTW becoming a practical similarity measure between curves.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Computational Geometry, Curve Similarity, Fréchet distance, Dynamic Time Warping, Continuous Dynamic Time Warping

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.22

Related Version *Full Version:* <https://arxiv.org/abs/2203.04531>

Funding *André Nusser:* Part of this research was conducted while the author was at Saarbrücken Graduate School of Computer Science and Max Planck Institute for Informatics. The author is supported by the VILLUM Foundation grant 16582.

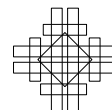
1 Introduction

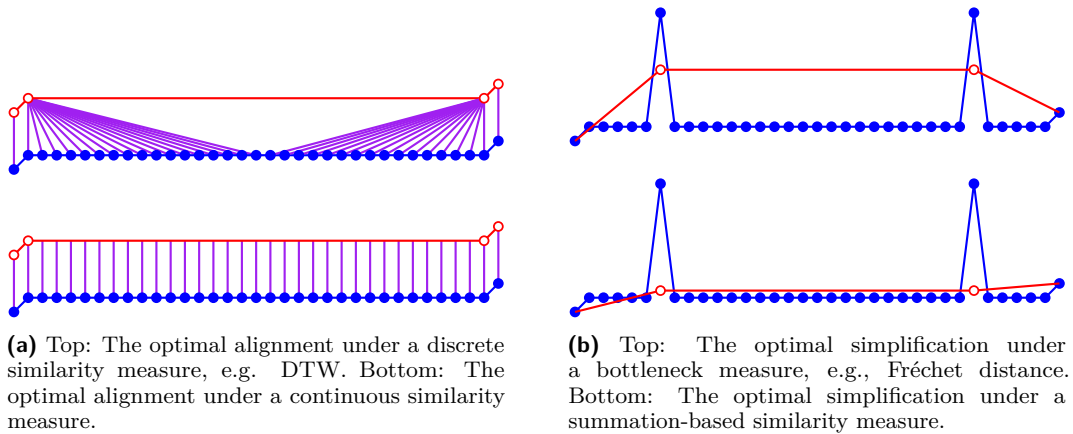
Time series data arises from many sources, such as financial markets [39], seismology [43], electrocardiography [5] and epidemiology [7]. Domain-specific questions can often be answered by analysing these time series. A common way of analysing time series is by finding similarities. Computing similarities is also a fundamental building block for other analyses, such as clustering, classification, or simplification. There are numerous similarity measures considered in literature [4, 19, 23, 26, 37, 40], many of which are application dependent.



© Kevin Buchin, André Nusser, and Sampson Wong;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 22; pp. 22:1–22:16
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



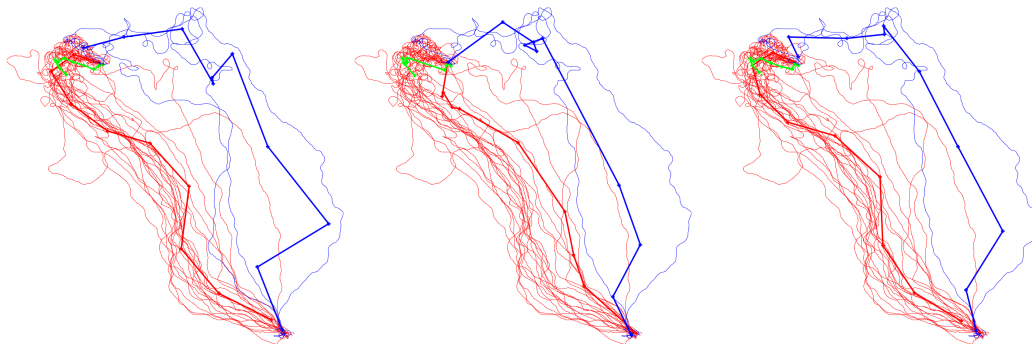


■ **Figure 1** Issues with discrete (left) and bottleneck (right) measures as opposed to continuous, summed measures.

Dynamic Time Warping (DTW) is arguably the most popular similarity measure for time series, and is widely used across multiple communities [2, 24, 30, 32, 33, 34, 38, 41, 42]. Under DTW, a minimum cost discrete alignment is computed between a pair of time series. A discrete alignment is a sequence of pairs of points, subject to the following four conditions: (i) the first pair is the first sample from both time series, (ii) the last pair is the last sample from both time series, (iii) each sample must appear in some pair in the alignment, and (iv) the alignment must be a monotonically increasing sequence for both time series. The cost of a discrete alignment, under DTW, is the sum of the distances between aligned points. A drawback of a similarity measure with a discrete alignment is that it is sensitive to the sampling rates of the time series. As such, DTW is a poor measure of similarity between a time series with a high sampling rate and a time series with a low sampling rate. For such cases, it is more appropriate to use a similarity measure with a continuous alignment. In Figure 1a, we provide a visual comparison of a discrete alignment versus a continuous alignment, for time series with vastly different sampling rates.

The Fréchet distance is a similarity measure that has gained popularity, especially in the theory community [3, 13, 20, 36]. To apply the Fréchet distance to a time series, we linearly interpolate between sampled points to obtain a continuous one-dimensional polygonal curve. Under the Fréchet distance, a minimum cost continuous alignment is computed between the pair of curves. A continuous alignment is a simultaneous traversal of the pair of curves that satisfies the same four conditions as previously stated for DTW. The cost of a continuous alignment, under the Fréchet distance, is the maximum distance between a pair of points in the alignment. The Fréchet distance is a bottleneck measure in that it only measures the maximum distance between aligned points. As a result, the drawback of the Fréchet distance is that it is sensitive to outliers. For such cases, a summation-based similarity measure is significantly more robust. In Figure 1b, we illustrate a high complexity curve, and its low complexity “simplification” that is the most similar to the original curve, under either a bottleneck or summation similarity measure. The simplified curve under the Fréchet distance is sensitive to and drawn towards its outlier points.

Continuous Dynamic Time Warping (CDTW) is a recently proposed alternative that does not exhibit the aforementioned drawbacks. It obtains the best of both worlds by combining the continuous nature of the Fréchet distance with the summation of DTW. CDTW was first introduced by Buchin [17], where it was referred to as the average Fréchet distance.



■ **Figure 2** Clustering of the c17 pigeon’s trajectories under the DTW (left), Fréchet (middle), and CDTW (right) distances. Figures were provided by the authors of [9].

CDTW has also been referred to as the summed, or integral, Fréchet distance. CDTW is similar to the Fréchet distance in that a minimum cost continuous alignment is computed between the pair of curves. The cost of a continuous alignment, under CDTW, is the integral of the distances between pairs of points in the alignment. We provide a formal definition in Section 2. Other definitions were also given under the name CDTW [22, 35], see Section 1.1.

Compared to existing popular similarity measures, CDTW is robust to both the sampling rate of the time series and to its outliers. CDTW has been used in applications where this robustness is desirable. In Brakatsoulas et al. [8], the authors applied CDTW to map-matching of vehicular location data. The authors highlight two common errors in real-life vehicular data, that is, measurement errors and sampling errors. Measurement errors result in outliers whereas sampling errors cause discrepancies in sampling rates between input curves. Their experiments show an improvement in map-matching when using CDTW instead of the Fréchet distance. In a recent paper, Brankovic et al. [9] applied CDTW to clustering of bird migration data and handwritten character data. The authors used (k, ℓ) -center and medians clustering, where each of the k clusters has a (representative) center curve of complexity at most ℓ . Low complexity center curves are used to avoid overfitting. Compared to DTW and the Fréchet distance, Brankovic et al. [9] demonstrated that clustering under CDTW produced centers that were more visually similar to the expected center curve. Under DTW, the clustering quality deteriorated for small values of ℓ , whereas under the Fréchet distance, the clustering quality deteriorated in the presence of outliers.

Brankovic et al.’s [9] clustering of a pigeon data set [28] is shown in Figure 2. The Fréchet distance is paired with the center objective, whereas DTW and CDTW are paired with the medians objective. Under DTW (left), the discretisation artifacts are visible. The blue center curve is jagged and visually dissimilar to its associated input curves. Under the Fréchet distance (middle), the shortcoming of the bottleneck measure and objective is visible. The red center curve fails to capture the shape of its associated input curves, in particular, it misses the top-left “hook” appearing in its associated curves. Under CDTW (right), the center curves are smooth and visually similar to their associated curves.

Despite its advantages, the shortcoming of CDTW is that there is no exact algorithm for computing it, in polynomial time or otherwise. Heuristics were used to compute CDTW in the map-matching [8] and clustering [9] experiments. Maheshwari et al. [27] provided a $(1 + \varepsilon)$ -approximation algorithm for CDTW in $\mathcal{O}(\zeta^4 n^3 / \varepsilon^2)$ time, for curves of complexity n and spread ζ , where the spread is the ratio between the maximum and minimum interpoint

distances. Existing heuristic and approximation methods [8, 9, 27] use a sampled grid on top of the dynamic program for CDTW, introducing an inherent error that depends on the fineness of the sampled grid, which is reflected in the dependency on ζ in [27].

In this work, we present the first exact algorithm for computing CDTW for one-dimensional curves. Our algorithm runs in time $\mathcal{O}(n^5)$ for a pair of one-dimensional curves, each with complexity at most n . Unlike previous approaches, we avoid using a sampled grid and instead devise a propagation method that solves the dynamic program for CDTW exactly. In our propagation method, the main difficulty lies in bounding the total complexity of our propagated functions. Showing that CDTW can be computed in polynomial time fosters hope for faster polynomial time algorithms, which would add CDTW to the list of practical similarity measures for curves.

1.1 Related work

Algorithms for computing popular similarity measures, such as DTW and the Fréchet distance, are well studied. Vintsyuk [41] proposed Dynamic Time Warping as a similarity measure for time series, and provided a simple dynamic programming algorithm for computing the DTW distance that runs in $\mathcal{O}(n^2)$ time, see also [6]. Gold and Sharir [24] improved the upper bound for computing DTW to $\mathcal{O}(n^2/\log \log n)$. For the Fréchet distance, Alt and Godau [3] proposed an $\mathcal{O}(n^2 \log n)$ time algorithm for computing the Fréchet distance between a pair of curves. Buchin et al. [13] improved the upper bound for computing the Fréchet distance to $\mathcal{O}(n^2 \sqrt{\log n} (\log \log n)^{3/2})$. Assuming SETH, it has been shown that there is no strongly subquadratic time algorithm for computing the Fréchet distance or DTW [1, 10, 11, 12, 16].

Our definition of CDTW was originally proposed by Buchin [17], and has since been used in several experimental works [8, 9]. We give Buchin's [17] definition formally in Section 2. Other definitions under the name CDTW have also been considered. We briefly describe the main difference between these definitions and the one used in this paper.

To the best of our knowledge, the first continuous version of DTW was by Serra and Berthod [35]. The same definition was later used by Munich and Perona [31]. Although a continuous curve is used in their definition, the cost of the matching is still a discrete summation of distances to sampled points. Our definition uses a continuous summation (i.e. integration) of distances between all points on the curves, and therefore, is more robust to discrepancies in sampling rate. Efrat et al. [22] proposed a continuous version of DTW that uses integration. However, their integral is defined in a significantly different way to ours. Their formulation minimises the change of the alignment and not the distance between aligned points. Thus, their measure is translational invariant and designed to compare the shapes of curves irrespective of their absolute positions in space.

2 Preliminaries

We use $[n]$ to denote the set $\{1, \dots, n\}$. To continuously measure the similarity of time series, we linearly interpolate between sampled points to obtain a one-dimensional polygonal curve. A one-dimensional polygonal curve P of complexity n is given by a sequence of vertices, $p_1, \dots, p_n \in \mathbb{R}$, connected in order by line segments. Furthermore, let $\|\cdot\|$ be the norm in the one-dimensional space \mathbb{R} . In higher dimensions, the Euclidean \mathcal{L}_2 norm is the most commonly used norm, but other norms such as \mathcal{L}_1 and \mathcal{L}_∞ may be used.

Consider a pair of one-dimensional polygonal curves $P = p_1, \dots, p_n$ and $Q = q_1, \dots, q_m$. Let $\Delta(n, m)$ be the set of all sequences of pairs of integers $(x_1, y_1), \dots, (x_k, y_k)$ satisfying $(x_1, y_1) = (1, 1)$, $(x_k, y_k) = (n, m)$ and $(x_{i+1}, y_{i+1}) \in \{(x_i + 1, y_i), (x_i, y_i + 1), (x_i + 1, y_i + 1)\}$.

The DTW distance between P and Q is defined as

$$d_{DTW}(P, Q) = \min_{\alpha \in \Delta(n, m)} \sum_{(x, y) \in \alpha} \|p_x - q_y\|.$$

The discrete Fréchet distance between P and Q is defined as

$$d_{dF}(P, Q) = \min_{\alpha \in \Delta(n, m)} \max_{(x, y) \in \alpha} \|p_x - q_y\|.$$

Let p and q be the total arc lengths of P and Q respectively. Define the parametrised curve $\{P(z) : z \in [0, p]\}$ to be the one-dimensional curve P parametrised by its arc length. In other words, $P(z)$ is a piecewise linear function so that the arc length of the subcurve from $P(0)$ to $P(z)$ is z . Define $\{Q(z) : z \in [0, q]\}$ analogously. Let $\Gamma(p)$ be the set of all continuous and non-decreasing functions $\alpha : [0, 1] \rightarrow [0, p]$ satisfying $\alpha(0) = 0$ and $\alpha(1) = p$. Let $\Gamma(p, q) = \Gamma(p) \times \Gamma(q)$. The continuous Fréchet distance between P and Q is defined as

$$d_F(P, Q) = \inf_{(\alpha, \beta) \in \Gamma(p, q)} \max_{z \in [0, 1]} \|P(\alpha(z)) - Q(\beta(z))\|,$$

The CDTW distance between P and Q is defined as

$$d_{CDTW}(P, Q) = \inf_{(\alpha, \beta) \in \Gamma(p, q)} \int_0^1 \|P(\alpha(z)) - Q(\beta(z))\| \cdot \|\alpha'(z) + \beta'(z)\| \cdot dz.$$

For the definition of CDTW, we additionally require that α and β are differentiable. The original intuition behind $d_{CDTW}(P, Q)$ is that it is a line integral in the parameter space, which we will define in Section 2.1. The term $\|\alpha'(z) + \beta'(z)\|$ implies that we are using the \mathcal{L}_1 metric in the parameter space, but other norms have also been considered [26, 27].

2.1 Parameter space under CDTW

The parameter space under CDTW is analogous to the free space diagram under the continuous Fréchet distance. Similar to previous work [17, 26, 27], we transform the problem of computing CDTW into the problem of computing a line integral in the parameter space.

Recall that the total arc lengths of P and Q are p and q respectively. The parameter space is defined to be the rectangular region $R = [0, p] \times [0, q]$ in \mathbb{R}^2 . The region is imbued with a metric $\|\cdot\|_R$. The \mathcal{L}_1 , \mathcal{L}_2 and \mathcal{L}_∞ norms have all been considered, but \mathcal{L}_1 is the preferred metric as it is the easiest to work with [26, 27]. At every point $(x, y) \in R$ we define the height of the point to be $h(x, y) = \|P(x) - Q(y)\|$.

Next, we provide the line integral formulation of d_{CDTW} , which is the original motivation behind its definition. To make our line integral easier to work with, we parametrise our line integral path γ in terms of its \mathcal{L}_1 arc length in R . The following lemma is a consequence of Section 6.2 in [17]. We provide a proof sketch of the result for the sake of self-containment.

► **Lemma 1.**

$$d_{CDTW}(P, Q) = \inf_{\gamma \in \Psi(p, q)} \int_0^{p+q} h(\gamma(z)) \cdot dz,$$

where $\Psi(p, q)$ is the set of all functions $\gamma : [0, p + q] \rightarrow R$ satisfying $\gamma(0) = (0, 0)$, $\gamma(p + q) = (p, q)$, γ is differentiable and non-decreasing in both x - and y -coordinates, and $\|\gamma'(z)\|_R = 1$.

Proof (Sketch). We provide a full proof in [15]. We summarise the main steps. Recall that the formula for CDTW is $\inf_{(\alpha, \beta) \in \Gamma(p, q)} \int_0^1 \|P(\alpha(z)) - Q(\beta(z))\| \cdot \|\alpha'(z) + \beta'(z)\| \cdot dz$. If we define $\gamma(t) = (\alpha(t), \beta(t))$, then the first term of the integrand is equal to $h(\gamma(t))$. Next, we reparametrise $\gamma(t)$ in terms of its \mathcal{L}_1 arc length in R . For our reparametrised γ , we get $\|\alpha'(z) + \beta'(z)\| = 1$, so the second term of the integrand is equal to 1. Finally, we prove that the parameter z ranges from 0 to $p + q$, and note that $\gamma(0) = (0, 0)$, $\gamma(p + q) = (p, q)$, γ is differentiable and non-decreasing, and $\|\gamma'(z)\|_R = 1$. This gives us the stated formula. ◀

2.2 Properties of the parameter space

Before providing the algorithm for minimising our line integral, we first provide some structural insights of our parameter space $R = [0, p] \times [0, q]$. Recall that $P : [0, p] \rightarrow \mathbb{R}$ maps points on the x -axis of R to points on the one-dimensional curve P , and analogously for Q and the y -axis. Hence, each point $(x, y) \in R$ is associated with a pair of points $P(x)$ and $Q(y)$, so that the height function $h(x, y) = \|P(x) - Q(y)\|$ is simply the distance between the associated pair of points. Divide the x -axis of R into $n - 1$ segments that are associated with the $n - 1$ segments $p_1p_2, \dots, p_{n-1}p_n$ of P . Divide the y -axis of R into $m - 1$ segments analogously. In this way, we divide R into $(n - 1)(m - 1)$ cells, which we label as follows:

► **Definition 2 (cell).** *Cell (i, j) is the region of the parameter space associated with segment $p_i p_{i+1}$ on the x -axis, and $q_j q_{j+1}$ on the y -axis, where $i \in [n - 1]$ and $j \in [m - 1]$.*

For points (x, y) restricted to a single cell (i, j) , the functions $P(x)$ and $Q(y)$ are linear. Hence, $P(x) - Q(y)$ is also linear, so $h(x, y) = \|P(x) - Q(y)\|$ is a piecewise linear surface with at most two pieces. If $h(x, y)$ consists of two linear surface pieces, the boundary of these two pieces is along a segment where $h(x, y) = 0$. Since we are working with one-dimensional curves, we have two cases for the relative directions of the vectors $\overrightarrow{p_i p_{i+1}}$ and $\overrightarrow{q_j q_{j+1}}$. If the vectors are in the same direction, since $\overrightarrow{p_i p_{i+1}}$ and $\overrightarrow{q_j q_{j+1}}$ are both parametrised by their arc lengths, they must be travelling in the same direction and at the same rate. Therefore, the line satisfying $h(x, y) = 0$ has slope 1 in R . Using a similar argument, if $\overrightarrow{p_i p_{i+1}}$ and $\overrightarrow{q_j q_{j+1}}$ are in opposite direction, then the line satisfying $h(x, y) = 0$ has slope -1 in R .

The line with zero height and slope 1 will play an important role in our algorithm. We call these lines valleys. If a path γ travels along a valley, the line integral accumulates zero cost as long as it remains on the valley, since the valley has zero height.

► **Definition 3 (valley).** *In a cell, the set of points (x, y) satisfying $h(x, y) = 0$ forms a line, moreover, the line has slope 1 or -1 . If the line has slope 1, we call it a valley.*

3 Algorithm

Our approach is a dynamic programming algorithm over the cells in the parameter space, which we defined in Section 2.1. To the best of our knowledge, all the existing approximation algorithms and heuristics [8, 9, 26] use a dynamic programming approach, or simply reduce the problem to a shortest path computation [27]. Next, we highlight the key difference between our approach and previous approaches.

In previous algorithms, sampling is used along cell boundaries to obtain a discrete set of grid points. Then, the optimal path between the discrete set of grid points is computed. The shortcoming of previous approaches is that an inherent error is introduced by the grid points, where the error depends on the fineness of the grid that is used.

In our algorithm, we consider all points along cell boundaries, not just a discrete subset. However, this introduces a challenge whereby we need to compute optimal paths between continuous segments of points. To overcome this obstacle, we devise a new method of propagating continuous functions across a cell. The main difficulty in analysing the running time of our algorithm lies in bounding the total complexity of the propagated continuous functions, across all cells in the dynamic program.

Our improvement over previous approaches is in many ways similar to previous algorithms for the weighted region problem [29], and the partial curve matching problem [14]. In all three problems, a line integral is minimised over a given terrain, and an optimal path is computed instead of relying on a sampled grid. However, our problem differs from that of [29] and [14] in two important ways. First, in both [29] and [14], the terrain is a piecewise constant function, whereas in our problem, the terrain is a piecewise linear function. Second, our main difficulty is bounding the complexity of the propagated functions. In [29], a different technique is used that does not propagate functions. In [14], the propagated functions are concave, piecewise linear and their complexities remain relatively low. In our algorithm, the propagated functions are piecewise quadratic and their complexities increase at a much higher, albeit bounded, rate.

3.1 Dynamic program

Our dynamic program is performed with respect to the following cost function.

► **Definition 4** (cost function). *Let $(x, y) \in R$, we define*

$$\text{cost}(x, y) = \inf_{\gamma \in \Psi(x, y)} \int_0^{x+y} h(\gamma(z)) \cdot dz.$$

Recall from Lemma 1 that $d_{CDTW}(P, Q) = \inf_{\gamma \in \Psi(p, q)} \int_0^{p+q} h(\gamma(z)) \cdot dz$, which implies that $\text{cost}(p, q) = d_{CDTW}(P, Q)$. Another way of interpreting Definition 4 is that $\text{cost}(x, y)$ is equal to $d_{CDTW}(P_x, Q_y)$, where P_x is the subcurve from $P(0)$ to $P(x)$, and Q_y is the subcurve from $Q(0)$ to $Q(y)$.

Recall from Section 2.2 that the parameter space is divided into $(n-1)(m-1)$ cells. Our dynamic program solves cells one at a time, starting from the bottom left cell and working towards the top right cell. A cell is considered solved if we have computed the cost of every point on the boundary of the cell. Once we solve the top right cell of R , we obtain the cost of the top right corner of R , which is $\text{cost}(p, q) = d_{CDTW}(P, Q)$, and we are done.

In the base case, we compute the cost of all points lying on the lines $x = 0$ and $y = 0$. Note that if $x = 0$ or $y = 0$, then the function $\text{cost}(x, y)$ is simply a function in terms of y or x respectively. In general, the function along any cell boundary – top, bottom, left or right – is a univariate function in terms of either x or y . We call these boundary cost functions.

► **Definition 5** (boundary cost function). *A boundary cost function is $\text{cost}(x, y)$, but restricted to a top, bottom, left or right boundary of a cell. If it is restricted to a top or bottom (resp. left or right) boundary, the boundary cost function is univariate in terms of x (resp. y).*

In the propagation step, we use induction to solve the cell (i, j) for all $1 \leq i \leq n-1$ and $1 \leq j \leq m-1$. We assume the base case. We also assume as an inductive hypothesis that, if $i \geq 2$, then the cell $(i-1, j)$ is already solved, and if $j \geq 2$, then the cell $(i, j-1)$ is already solved. Our assumptions ensure that we receive as input the boundary cost function along the bottom and left boundaries of the cell (i, j) . In other words, we use the boundary cost functions along the input boundaries to compute the boundary cost functions along the output boundaries.

► **Definition 6** (input/output boundary). *The input boundaries of a cell are its bottom and left boundaries. The output boundaries of a cell are its top and right boundaries.*

We provide details of the base case in Section 3.2, and the propagation step in Section 3.3.

3.2 Base case

The base case is to compute the cost of all points along the x -axis. The y -axis can be handled analogously. Recall that $cost(x, y) = \inf_{\gamma \in \Psi(x, y)} \int_0^{x+y} h(\gamma(z)) \cdot dz$. Therefore, for points $(x, 0)$ on the x -axis, we have $cost(x, 0) = \inf_{\gamma \in \Psi(x, 0)} \int_0^x h(\gamma(z)) \cdot dz$. Since $\gamma(z)$ is non-decreasing in x - and y -coordinates, and $\|\gamma'(z)\| = 1$, we must have that $\gamma'(z) = (1, 0)$. By integrating from 0 to z , we get $\gamma(z) = (z, 0)$, which implies that $cost(x, 0) = \int_0^x h(z, 0) \cdot dz$.

Consider, for $1 \leq i \leq n-1$, the bottom boundary of the cell $(i, 1)$. The height function $h(z)$ is a piecewise linear function with at most two pieces, so its integral $cost(x, 0) = \int_0^x h(z, 0) \cdot dz$ is a continuous piecewise quadratic function with at most two pieces. Similarly, since the height function along $x = 0$ is a piecewise linear function with at most $2(n-1)$ pieces, the boundary cost function along $x = 0$ is a continuous piecewise quadratic function with at most $2(n-1)$ pieces. For boundaries not necessarily on the x - or y -axis, we claim that the boundary cost function is still a continuous piecewise-quadratic function.

► **Lemma 7.** *The boundary cost function is a continuous piecewise-quadratic function.*

We defer the proof of Lemma 7 to the full version of this paper [15]. Although the boundary cost function has at most two pieces for cell boundaries on the x - or y -axis, in the general case it may have more than two pieces. As previously stated, the main difficulty in bounding our running time analysis in Section 3.4 is to bound complexities of the boundary cost functions.

3.3 Propagation step

First, we define optimal paths in the parameter space. We use optimal paths to propagate the boundary cost functions across cells in the parameter space. Note that the second part of Definition 8 is a technical detail to ensure the uniqueness of optimal paths. Intuitively, the optimal path from s to t is the path minimising the path integral, and if there are multiple such paths, the optimal path is the one with maximum y -coordinate.

► **Definition 8** (optimal path). *Given $t = (x_t, y_t) \in R$, its optimal path is a path $\gamma \in \Psi(x_t, y_t)$ minimising the integral $\int_0^{x_t+y_t} h(\gamma(z)) \cdot dz$. If there are multiple such curves that minimise the integral, the optimal path is the one with maximum y -coordinate (or formally, the one with maximum integral of its y -coordinates).*

Suppose t is on the output boundary of the cell (i, j) . Consider the optimal path γ that starts at $(0, 0)$ and ends at t . Let s be the first point where γ enters the cell (i, j) . We consider the subpath from s to t , which is entirely contained in the cell (i, j) . In the next lemma, we show that the shape of the subpath from s to t is restricted, in particular, there are only three types of paths that we need to consider.

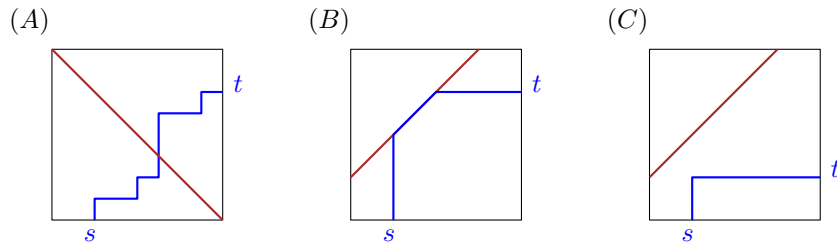
► **Lemma 9.** *Let t be a point on the output boundary of a cell. Let s be the first point where the optimal path to t enters the cell. There are only three types of paths from s to t :*

(A) *The segments of the cell are in opposite directions. Then all paths between s and t have the same cost.*

(B) The segments of the cell are in the same direction and the optimal path travels towards the valley, then along the valley, then away from the valley.

(C) The segments of the cell are in the same direction and the optimal path travels towards the valley, then away from the valley.

For an illustration of these three types of paths, see Figure 3.



■ **Figure 3** The three types of optimal paths through a cell.

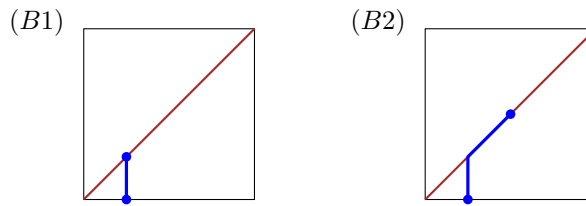
Proof (Sketch). A similar proof can be found in Lemma 4 of Maheswari et al. [27]. Nonetheless, due to slight differences, we provide a full proof in the full version [15]. Specifically, we consider one-dimensional curves, and use the \mathcal{L}_1 norm in parameter space to obtain a significantly stronger statement for type (A) paths.

We summarise the main steps here. Define γ_1 to be an optimal path to s , followed by any path from s to t . Define γ_2 to be an optimal path to s , followed by either a type (A), (B) or (C) path from s to t . If the segments are in opposite directions, we use a type (A) path, whereas if the segments are in the same direction, we use either a type (B) or type (C) path. The main step is to show that $h(\gamma_1(z)) \geq h(\gamma_2(z))$, as this would imply that γ_2 is an optimal path from s to t . In fact, if the segments are in the opposite directions, we get that $h(\gamma_1(z)) = h(\gamma_2(z))$, implying that all type (A) paths from s to t have the same cost. ◀

We leverage Lemma 9 to propagate the boundary cost function from the input boundaries to the output boundaries of a cell. We provide an outline of our propagation procedure in one of the three cases, that is, for type (B) paths. These paths are the most interesting to analyse, and looking at this special case provides us with some intuition for the other cases. For type (B) paths, we compute the cost function along the output boundary in three consecutive steps. We first list the steps, then we describe the steps in detail.

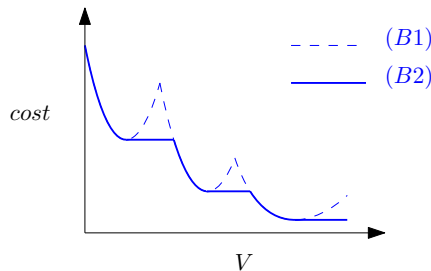
1. We compute the cost function along the valley in a restricted sense.
2. We compute the cost function along the valley in general.
3. We compute the cost function along the output boundary.

In the first step, we restrict our attention only to paths that travel from the input boundary towards the valley. This is the first segment in the type (B) path as defined in Lemma 9. We call this first segment a type (B1) path, see Figure 4. Define the type (B1) cost function to be the cost function along the valley if we can only use type (B1) paths from the input boundary to the valley. The type (B1) cost function is simply the cost function along the bottom or left boundary plus the integral of the height function along the type (B1) path. The height function along the type (B1) path is a linear function, so the integral is a quadratic function. To obtain the type (B1) cost function, we add the quadratic function for the type (B1) path to the cost function along an input boundary. We combine the type (B1) cost functions along the bottom and the left boundaries by taking their lower envelope.



■ **Figure 4** The type (B1) and type (B2) paths from the bottom boundary to the valley.

In the second step, we compute the cost function along the valley in general. It suffices to consider paths that travel from the input boundary towards the valley, and then travel along the valley. This path is the first two segments in a type (B) path as defined in Lemma 9. We call these first two segments a type (B2) path, see Figure 4. Since the height function is zero along the valley, if we can reach a valley point with a particular cost with a type (B1) path, then we can reach all points on the valley above and to the right of it with a type (B2) path with the same cost. Therefore, the type (B2) cost function is the cumulative minimum of the type (B1) cost function, see Figure 5. Note that the type (B2) cost function may have more quadratic pieces than the type (B1) cost function. For example, in Figure 5, the type (B2) cost function has twice as many quadratic pieces as the type (B1) cost function, since each quadratic piece in the type (B1) cost function splits into two quadratic pieces in the type (B2) cost function – the original quadratic piece plus an additional horizontal piece.



■ **Figure 5** The type (B2) cost function plotted over its position along the valley V . The type (B2) cost function is the cumulative minimum of the type (B1) cost function.

In the third step, we compute the cost function along the output boundary, given the type (B2) cost function along the valley. A type (B) path is a type (B2) path appended with a horizontal or vertical path from the valley to the boundary. The height function of the appended path is a linear function, so its integral is a quadratic function. We add this quadratic function to the type (B2) function along the valley to obtain the output function. This completes the description of the propagation step in the type (B) paths case.

Using a similar approach, we can compute the cost function along the output boundary in the type (A) and type (C) paths as well. The propagation procedure differs slightly for each of the three path types, for details see the full version [15]. Recall that due to the second step of the type (B) propagation, each quadratic piece along the input boundary may propagate to up to two pieces along the output boundary. In general, we claim that each quadratic piece along the input boundary propagates to at most a constant number of pieces along the output boundary. Moreover, given a single input quadratic piece, this constant number of output quadratic pieces can be computed in constant time.

► **Lemma 10.** *Each quadratic piece in the input boundary cost function propagates to at most a constant number of pieces along the output boundary. Propagating a quadratic piece takes constant time.*

We prove Lemma 10 in the full version [15]. We can now state our propagation step in general. Divide the input boundaries into subsegments, so that for each subsegment, the cost function along that subsegment is a single quadratic piece. Apply Lemma 10 to a subsegment to compute in constant time a piecewise quadratic cost function along the output boundary. Apply this process to all subsegments to obtain a set of piecewise quadratic cost functions along the output boundary. Combine these cost functions by taking their lower envelope. Return this lower envelope as the boundary cost function along the output boundary. This completes the statement of our propagation step. Its correctness follows from construction.

3.4 Running time analysis

We start the section with a useful lemma. Essentially the same result is stated without proof as Observation 3.3 in [14]. For the sake of completeness, we provide a proof sketch.

► **Lemma 11.** *Let γ_1, γ_2 be two optimal paths. These paths cannot cross, i.e., there are no z_1, z_2 such that $\gamma_1(z_1)$ is below $\gamma_2(z_1)$ and $\gamma_1(z_2)$ is above $\gamma_2(z_2)$.*

Proof (Sketch). We provide a full proof in [15]. We summarise the main steps. Let u be the first point where a pair of optimal paths crosses. We show that the crossing paths, up to u , must have been identical, so the paths cannot cross at u . ◀

Define N to be the total number of quadratic pieces in the boundary cost functions over all boundaries of all cells. We will show that the running time of our algorithm is $\mathcal{O}(N)$.

► **Lemma 12.** *The running time of our dynamic programming algorithm is $\mathcal{O}(N)$.*

Proof. The running time of the dynamic program is dominated by the propagation step. Let $I_{i,j}$ denote the input boundaries of the cell (i, j) . Let $|I_{i,j}|$ denote the number of quadratic functions in the input boundary cost function. By Lemma 10, each piece only propagates to a constant number of new pieces along the output boundary, and these pieces can be computed in constant time. The final piecewise quadratic function is the lower envelope of all the new pieces, of which there are $\mathcal{O}(|I_{i,j}|)$ many.

We use Lemma 11 to speed up the computation of the lower envelope, so that this step takes only $\mathcal{O}(|I_{i,j}|)$ time. Since optimal paths do not cross, it implies that the new pieces along the output boundary appear in the same order as their input pieces. We perform the propagation in order of the input pieces. We maintain the lower envelope of the new pieces in a stack. For each newly propagated piece, we remove the suffix that is dominated by the new piece and then add the new piece to the stack. Since each quadratic piece can be added to the stack at most once, and removed from the stack at most once, the entire operation takes $\mathcal{O}(|I_{i,j}|)$ time. Summing over all cells, we obtain an overall running time of $\mathcal{O}(N)$. ◀

Note that Lemma 12 does not yet guarantee that our algorithm runs in polynomial time as we additionally need to bound N by a polynomial. Lemma 10 is of limited help. The lemma states that each piece on the input boundary propagates to at most a constant number of pieces on the output boundary. Recall that in Section 3.3, we illustrated a type (B) path that resulted in an output boundary having twice as many quadratic pieces as its input boundary. The doubling occurred in the second step of the propagation of type (B) paths, see Figure 5. If this doubling behaviour were to occur for all our cells in our dynamic

program, we would get up to $N = \Omega(2^{n+m})$ quadratic pieces in the worst case, where n and m are the complexities of the polygonal curves P and Q . To obtain a polynomial running time, we show that although this doubling behaviour may occur, it does not occur *too often*.

3.5 Bounding the cost function's complexity

Our bound comes in two parts. First, we subdivide the boundaries in the parameter space into subsegments and show, in Lemma 13, that there are $\mathcal{O}((n+m)^3)$ subsegments in total. Second, in Lemma 15, we show that each subsegment has at most $\mathcal{O}((n+m)^2)$ quadratic pieces. Putting this together in Theorem 16 gives $N = \mathcal{O}((n+m)^5)$.

We first define the $\mathcal{O}((n+m)^3)$ subsegments. The intuition behind the subsegments is that for any two points on the subsegment, the optimal path to either of those two points is structurally similar. We can deform one of the optimal paths to the other without passing through any cell corner, or any points where a valley meets a boundary.

Formally, define A_k to be the union of the input boundaries of the cells (i, j) such that $i + j = k$. Alternatively, A_k is the union of the output boundaries of the cells (i, j) such that $i + j + 1 = k$. Next, construct the partition $A_k := \{A_{k,1}, A_{k,2}, \dots, A_{k,L}\}$ of A_k into subsegments. Define the subsegment $A_{k,\ell}$ to be the segment between the ℓ^{th} and $(\ell + 1)^{\text{th}}$ critical point along A_k . We define a critical point to be either (i) a cell corner, (ii) a point where the valley meets the boundary, or (iii) a point where the optimal path switches from passing through a subsegment $A_{k-1,\ell'}$ to a different subsegment $A_{k-1,\ell''}$.

Let $|A_k|$ denote the number of piecewise quadratic cost functions $A_{k,\ell}$ along A_k . Let $|A_{k,\ell}|$ denote the number of pieces in the piecewise quadratic cost function along the subsegment $A_{k,\ell}$. Thus, we can rewrite the total number of quadratic functions N as:

$$N = \sum_{k=2}^{n+m-1} \sum_{\ell=1}^{|A_k|} |A_{k,\ell}|.$$

We first show that the number of subsegments $|A_k|$ is bounded by $\mathcal{O}(k^2)$ and then proceed to show that $|A_{k,\ell}|$ is bounded by $\mathcal{O}(k^2)$ for all k, ℓ .

► **Lemma 13.** *For any $k \in [n+m]$, we have $|A_k| \leq 2k^2$.*

Proof. We prove the lemma by induction. Since the cell $(1, 1)$ has at most one valley, and since the input boundary A_2 has one cell corner, we have $|A_2| \leq 3$. For the inductive step, note that there are at most $2k$ cell corners on A_k , and there are at most k points where a valley meets a boundary on A_k . By the inductive hypothesis, there are at most $2(k-1)^2$ subsegments on A_{k-1} . And as optimal paths do not cross by Lemma 11, each subsegment of A_{k-1} contributes at most once to the optimal path switching from one subsegment to a different one on A_k . Thus, for $k \geq 3$, we obtain $|A_k| \leq 2(k-1)^2 + 2k + k + 1 = 2k^2 - k + 3 \leq 2k^2$. ◀

Next, we show that $|A_{k,\ell}|$ is bounded by $\mathcal{O}(k^2)$ for all k, ℓ . We proceed by induction. Recall that, due to the third type of critical point, all optimal paths to $A_{k,\ell}$ pass through the same subsegment of A_{k-1} , namely $A_{k-1,\ell'}$ for some ℓ' . Our approach is to assume the inductive hypothesis for $|A_{k-1,\ell'}|$, and bound $|A_{k,\ell}|$ relative to $|A_{k-1,\ell'}|$. We already have a bound of this form, specifically, Lemma 10 implies that $|A_{k,\ell}| \leq c \cdot |A_{k-1,\ell'}|$, for some constant $c > 1$. Unfortunately, this bound does not rule out an exponential growth in the cost function complexity. We instead prove the following improved bound:

► **Lemma 14.** *Let $|A_{k,\ell}|$ be a subsegment on A_k , and suppose all optimal paths to $|A_{k,\ell}|$ pass through subsegment $|A_{k-1,\ell'}|$ on A_{k-1} . Then*

$$\begin{aligned} |A_{k,\ell}| &\leq |A_{k-1,\ell'}| + D(A_{k-1,\ell'}), \\ D(A_{k,\ell}) &\leq D(A_{k-1,\ell'}) + 1, \end{aligned}$$

where $D(\cdot)$ counts, for a given subsegment, the number of distinct pairs (a, b) over all quadratics $ax^2 + bx + c$ in the boundary cost function for that subsegment.

We prove Lemma 14 in [15]. The lemma obtains a polynomial bound on the growth of the number of quadratic pieces by showing, along the way, a polynomial bound on the growth of the number of distinct (a, b) pairs over the quadratics $ax^2 + bx + c$.

As we consider this lemma to be one of the main technical contributions of the paper, we will briefly outline its intuition. It is helpful for us to revisit the doubling behaviour of type (B) paths. Recall that in our example in Figure 5, we may have $|A_{k,\ell}| = 2|A_{k-1,\ell'}|$. This doubling behaviour does not contradict Lemma 14, so long as all quadratic functions along $A_{k-1,\ell'}$ have distinct (a, b) pairs. In fact, for $|A_{k,\ell}| = 2|A_{k-1,\ell'}|$ to occur, each quadratic function in $|A_{k-1,\ell'}|$ must create a new horizontal piece in the cumulative minimum step. But for any two quadratic functions with the same (a, b) pair, only one of them can create a new horizontal piece, since the horizontal piece starts at the x -coordinate $-\frac{b}{2a}$. Therefore, we must have had that all quadratic functions along $A_{k-1,\ell'}$ have distinct (a, b) pairs. In [15], we generalise this argument and prove $|A_{k,\ell}| \leq |A_{k-1,\ell'}| + D(A_{k-1,\ell'})$.

We perform a similar analysis in the special case of type (B) paths to give the intuition behind $D(A_{k,\ell}) \leq D(A_{k-1,\ell'}) + 1$. For type (B) paths, the number of distinct (a, b) pairs changes only in the cumulative minimum step. All pieces along $A_{k,\ell}$ can either be mapped to a piece along $A_{k-1,\ell'}$, or it is a new horizontal piece. However, all new horizontal pieces have an (a, b) pair of $(0, 0)$, so the number of distinct (a, b) pairs increases by only one. For the full proof of Lemma 14 for all three path types, refer to the full version [15].

With Lemma 14 in mind, we can now prove a bound on $|A_{k,\ell}|$ by induction.

► **Lemma 15.** *For any $k \in [n + m]$ and $A_{k,\ell} \in A_k$ we have $|A_{k,\ell}| \leq k^2$.*

Proof. Note that in the base case $D(A_{2,\ell}) \leq 2$ and $|A_{2,\ell}| \leq 4$ for any $A_{2,\ell} \in A_2$. By Lemma 14, we get $D(A_{k,\ell}) \leq D(A_{k-1,\ell'}) + 1$, for some subsegment $A_{k-1,\ell'}$ on A_{k-1} . By a simple induction, we get $D(A_{k,\ell}) \leq k$ for any $k \in [n + m]$. Similarly, assuming $|A_{k-1,\ell}| \leq (k - 1)^2$ for any $A_{k-1,\ell} \in A_{k-1}$, we use Lemma 14 to inductively obtain $|A_{k,\ell}| \leq |A_{k-1,\ell'}| + D(A_{k-1,\ell'}) + 1 \leq |A_{k-1,\ell'}| + k \leq (k - 1)^2 + k - 1 + 1 \leq k^2$ for any $A_{k,\ell} \in A_k$. ◀

Using our lemmas, we can finally bound N , and thereby the overall running time.

► **Theorem 16.** *The Continuous Dynamic Time Warping distance between two 1-dimensional polygonal curves of length n and m , respectively, can be computed in time $\mathcal{O}((n + m)^5)$.*

Proof. Using Lemmas 13 and 15, we have

$$N = \sum_{k=2}^{n+m-1} \sum_{\ell=1}^{|A_k|} |A_{k,\ell}| \leq \sum_{k=2}^{n+m} \sum_{\ell=1}^{|A_k|} k^2 \leq \sum_{k=2}^{n+m} 2k^4 \leq 2(n + m)^5.$$

Thus, the overall running time of our algorithm is $\mathcal{O}((n + m)^5)$, by Lemma 12. ◀

4 Conclusion

We presented the first exact algorithm for computing CDTW of one-dimensional curves, which runs in polynomial time. Our main technical contribution is bounding the total complexity of the functions which the algorithm propagates, to bound the total running time of the algorithm. One direction for future work is to improve the upper bound on the total complexity of the propagated functions. Our $O(n^5)$ upper bound is pessimistic, for example, we do not know of a worst case instance. Another direction is to compute CDTW in higher dimensions. In two dimensions, the Euclidean \mathcal{L}_2 norm is the most commonly used norm, however, this is likely to result in algebraic issues similar to that for the weighted region problem [18]. One way to avoid these algebraic issues is to use a polyhedral norm, such as the \mathcal{L}_1 , \mathcal{L}_∞ , or an approximation of the \mathcal{L}_2 norm [21, 25]. This would result in an approximation algorithm similar to [27], but without a dependency on the spread.

References

- 1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pages 59–78. IEEE Computer Society, 2015.
- 2 Pankaj K. Agarwal, Kyle Fox, Jiangwei Pan, and Rex Ying. Approximating dynamic time warping and edit distance for a pair of point sequences. In Sándor P. Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry, SoCG 2016*, volume 51 of *LIPICs*, pages 6:1–6:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.
- 3 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.*, 5:75–91, 1995.
- 4 Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. Spatio-temporal data mining: A survey of problems and methods. *ACM Comput. Surv.*, 51(4):83:1–83:41, 2018.
- 5 Selcan Kaplan Berkaya, Alper Kursat Uysal, Efnan Sora Gunal, Semih Ergin, Serkan Gunal, and M Bilginer Gulmezoglu. A survey on ECG analysis. *Biomedical Signal Processing and Control*, 43:216–235, 2018.
- 6 Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop, Seattle, Washington, USA, July 1994. Technical Report WS-94-03*, pages 359–370. AAAI Press, 1994.
- 7 Krishnan Bhaskaran, Antonio Gasparrini, Shakoor Hajat, Liam Smeeth, and Ben Armstrong. Time series regression studies in environmental epidemiology. *International Journal of Epidemiology*, 42(4):1187–1195, 2013.
- 8 Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *Proceedings of the 31st International Conference on Very Large Data Bases, VLDB 2005*, pages 853–864. ACM, 2005.
- 9 Milutin Brankovic, Kevin Buchin, Koen Klaren, André Nusser, Aleksandr Popov, and Sampson Wong. (k, ℓ) -medians clustering of trajectories using continuous dynamic time warping. In *SIGSPATIAL '20: 28th International Conference on Advances in Geographic Information Systems*, pages 99–110. ACM, 2020.
- 10 Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014*, pages 661–670. IEEE Computer Society, 2014.
- 11 Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pages 79–97. IEEE Computer Society, 2015.
- 12 Karl Bringmann and Wolfgang Mulzer. Approximability of the discrete Fréchet distance. *J. Comput. Geom.*, 7(2):46–76, 2016.

- 13 Kevin Buchin, Maike Buchin, Wouter Meulemans, and Wolfgang Mulzer. Four soviets walk the dog: Improved bounds for computing the Fréchet distance. *Discret. Comput. Geom.*, 58(1):180–216, 2017.
- 14 Kevin Buchin, Maike Buchin, and Yusu Wang. Exact algorithms for partial curve matching via the fréchet distance. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009*, pages 645–654. SIAM, 2009.
- 15 Kevin Buchin, André Nusser, and Sampson Wong. Computing continuous dynamic time warping of time series in polynomial time. *CoRR*, abs/2203.04531, 2022.
- 16 Kevin Buchin, Tim Ophelders, and Bettina Speckmann. SETH says: Weak Fréchet distance is faster, but only if it is continuous and in one dimension. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, pages 2887–2901. SIAM, 2019.
- 17 Maike Buchin. *On the computability of the Fréchet distance between triangulated surfaces*. PhD thesis, Freie Universität Berlin, 2007.
- 18 Jean-Lou De Carufel, Carsten Grimm, Anil Maheshwari, Megan Owen, and Michiel H. M. Smid. A note on the unsolvability of the weighted region shortest path problem. *Comput. Geom.*, 47(7):724–727, 2014.
- 19 Ian R Cleasby, Ewan D Wakefield, Barbara J Morrissey, Thomas W Bodey, Steven C Votier, Stuart Bearhop, and Keith C Hamer. Using time-series similarity measures to compare animal movement trajectories in ecology. *Behavioral Ecology and Sociobiology*, 73(11):1–19, 2019.
- 20 Anne Driemel, Amer Krivosija, and Christian Sohler. Clustering time series under the Fréchet distance. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, pages 766–785. SIAM, 2016.
- 21 Richard M Dudley. Metric entropy of some classes of sets with differentiable boundaries. *Journal of Approximation Theory*, 10(3):227–236, 1974.
- 22 Alon Efrat, Quanfu Fan, and Suresh Venkatasubramanian. Curve matching, time warping, and light fields: New algorithms for computing similarity between curves. *J. Math. Imaging Vis.*, 27(3):203–216, 2007.
- 23 Philippe Esling and Carlos Agón. Time-series data mining. *ACM Comput. Surv.*, 45(1):12:1–12:34, 2012.
- 24 Omer Gold and Micha Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Trans. Algorithms*, 14(4):50:1–50:17, 2018.
- 25 Sarel Har-Peled and Mitchell Jones. Proof of Dudley’s convex approximation. *arXiv preprint*, 2019. [arXiv:1912.01977](https://arxiv.org/abs/1912.01977).
- 26 Koen Klaren. Continuous dynamic time warping for clustering curves. Master’s thesis, Eindhoven University of Technology, 2020.
- 27 Anil Maheshwari, Jörg-Rüdiger Sack, and Christian Scheffer. Approximating the integral Fréchet distance. *Comput. Geom.*, 70-71:13–30, 2018.
- 28 Jessica Meade, Dora Biro, and Tim Guilford. Homing pigeons develop local route stereotypy. *Proceedings of the Royal Society B: Biological Sciences*, 272(1558):17–23, 2005.
- 29 Joseph S. B. Mitchell and Christos H. Papadimitriou. The weighted region problem: Finding shortest paths through a weighted planar subdivision. *J. ACM*, 38(1):18–73, 1991.
- 30 Meinard Müller. Dynamic time warping. In *Information Retrieval for Music and Motion*, pages 69–84. Springer, 2007.
- 31 Mario E. Munich and Pietro Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Proceedings of the International Conference on Computer Vision, 1999*, pages 108–115. IEEE Computer Society, 1999.
- 32 Cory Myers, Lawrence Rabiner, and Aaron Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(6):623–635, 1980.

- 33 Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.
- 34 Pavel Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaiï at Manoa Honolulu, USA*, 855(1-23):40, 2008.
- 35 Bruno Serra and Marc Berthod. Subpixel contour matching using continuous dynamic programming. In *Conference on Computer Vision and Pattern Recognition, CVPR 1994*, pages 202–207. IEEE, 1994.
- 36 E. Sriraghavendra, K. Karthik, and Chiranjib Bhattacharyya. Fréchet distance based approach for searching online handwritten documents. In *9th International Conference on Document Analysis and Recognition, ICDAR 2007*, pages 461–465. IEEE Computer Society, 2007.
- 37 Yaguang Tao, Alan Both, Rodrigo I Silveira, Kevin Buchin, Stef Sijben, Ross S Purves, Patrick Laube, Dongliang Peng, Kevin Toohey, and Matt Duckham. A comparative analysis of trajectory similarity measures. *GIScience & Remote Sensing*, pages 1–27, 2021.
- 38 Charles C. Tappert, Ching Y. Suen, and Toru Wakahara. The state of the art in online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):787–808, 1990.
- 39 Stephen J Taylor. *Modelling financial time series*. World Scientific, 2008.
- 40 Kevin Toohey and Matt Duckham. Trajectory similarity measures. *ACM SIGSPATIAL Special*, 7(1):43–50, 2015.
- 41 Taras K Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics*, 4(1):52–57, 1968.
- 42 Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn J. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Min. Knowl. Discov.*, 26(2):275–309, 2013.
- 43 Öz Yilmaz. *Seismic data analysis: Processing, inversion, and interpretation of seismic data*. Society of exploration geophysicists, 2001.

Long Plane Trees

Sergio Cabello ✉ 

Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia
Faculty of Mathematics and Physics, University of Ljubljana, Slovenia

Michael Hoffmann ✉ 

Department of Computer Science, ETH Zürich, Switzerland

Katharina Klost ✉

Institut für Informatik, Freie Universität Berlin, Germany

Wolfgang Mulzer ✉ 

Institut für Informatik, Freie Universität Berlin, Germany

Josef Tkadlec ✉ 

Department of Mathematics, Harvard University, Cambridge, MA, USA

Abstract

In the *longest plane spanning tree* problem, we are given a finite planar point set \mathcal{P} , and our task is to find a plane (i.e., noncrossing) spanning tree T_{OPT} for \mathcal{P} with maximum total Euclidean edge length $|T_{\text{OPT}}|$. Despite more than two decades of research, it remains open if this problem is NP-hard. Thus, previous efforts have focused on polynomial-time algorithms that produce plane trees whose total edge length approximates $|T_{\text{OPT}}|$. The approximate trees in these algorithms all have small unweighted diameter, typically three or four. It is natural to ask whether this is a common feature of longest plane spanning trees, or an artifact of the specific approximation algorithms.

We provide three results to elucidate the interplay between the approximation guarantee and the unweighted diameter of the approximate trees. First, we describe a polynomial-time algorithm to construct a plane tree T_{ALG} with diameter at most four and $|T_{\text{ALG}}| \geq 0.546 \cdot |T_{\text{OPT}}|$. This constitutes a substantial improvement over the state of the art. Second, we show that a longest plane tree among those with diameter at most three can be found in polynomial time. Third, for any candidate diameter $d \geq 3$, we provide upper bounds on the approximation factor that can be achieved by a longest plane tree with diameter at most d (compared to a longest plane tree without constraints).

2012 ACM Subject Classification Theory of computation → Routing and network design problems; Theory of computation → Approximation algorithms analysis; Theory of computation → Computational geometry; Mathematics of computing → Trees

Keywords and phrases geometric network design, spanning trees, plane straight-line graphs, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.23

Related Version *Full Version*: <https://arxiv.org/abs/2101.00445> [11]

Funding *Sergio Cabello*: Supported by the Slovenian Research Agency (P1-0297, J1-9109, J1-8130, J1-8155, J1-1693, J1-2452).

Michael Hoffmann: Supported by the Swiss National Science Foundation within the collaborative DACH project *Arrangements and Drawings* as SNSF Project 200021E-171681.

Wolfgang Mulzer: Supported in part by ERC StG 757609.

1 Introduction

Geometric network design is a common and well-studied task in computational geometry and combinatorial optimization [18, 21, 24, 25]. In this family of problems, we are given a set \mathcal{P} of points, and our task is to connect \mathcal{P} into a (geometric) graph that has certain favorable properties. Not surprisingly, this general question has captivated the attention of researchers



© Sergio Cabello, Michael Hoffmann, Katharina Klost, Wolfgang Mulzer, and Josef Tkadlec;

licensed under Creative Commons License CC-BY 4.0

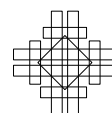
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 23; pp. 23:1–23:17

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



for a long time, and we can find countless variants, depending on which restrictions we put on the graph that connects \mathcal{P} and which criteria of this graph we would like to optimize. Typical graph classes of interest include matchings, paths, cycles, trees, or general *plane (noncrossing)* graphs, i.e., graphs, whose straight-line embedding on \mathcal{P} does not contain any edge crossings. Typical quality criteria include the total edge length [3, 15, 23, 28], the maximum length (bottleneck) edge [6, 17], the maximum degree [4, 12, 19, 31], the dilation [18, 26, 29], or the stabbing number [27, 33] of the graph. Many famous problems from computational geometry fall into this general setting. For example, if our goal is to minimize the total edge length, while restricting our considerations to paths, trees, or triangulations, respectively, we are faced with the venerable problems of finding an optimum TSP tour [21], a Euclidean minimum spanning tree [15], or a minimum weight triangulation [28] for \mathcal{P} . These three examples also illustrate the wide variety of complexity aspects that we may encounter in geometric network design problems: the Euclidean TSP is known to be NP-hard [30], but it admits a PTAS [3, 23]. On the other hand, it is possible to find a Euclidean minimum spanning tree for \mathcal{P} in polynomial time [15] (even though, curiously, the associated decision problem is not known to be solvable by a polynomial-time Turing machine, see, e.g., [9]). The minimum weight triangulation problem is also known to be NP-hard [28], but the existence of a PTAS is still open; however, a QPTAS is known [32].

In this work, we are interested in the interaction of two specific requirements for a geometric network design problem, namely the two objectives of obtaining a plane graph and of optimizing the total edge length. For the case that we want to *minimize* the total edge length of the resulting graph, these two goals are often in perfect harmony: the shortest Euclidean TSP tour and the shortest Euclidean minimum spanning tree are automatically plane, as can be seen by a simple application of the triangle inequality. In contrast, if our goal is to *maximize* the total edge length, while obtaining a plane graph, much less is known.

This family of problems was studied by Alon, Rajagopalan, and Suri [1], who considered computing a longest plane matching, a longest plane Hamiltonian path, and a longest plane spanning tree for a planar point set \mathcal{P} in general position. They conjectured that all three problems are NP-hard, but as far as we know, this is still open. The situation is similar for the problem of finding a *maximum weight triangulation* for \mathcal{P} : here, we have neither an NP-hardness proof nor a polynomial time algorithm [13]. If we omit the planarity condition, then the problem of finding a longest Hamiltonian path (the *geometric maximum TSP problem*) is known to be NP-hard in dimension three and above, while the two-dimensional case remains open [5]. On the other hand, we can find a longest (typically not plane) tree on \mathcal{P} in polynomial time, using classic greedy algorithms [14, Chapters 16.4, 23].

Longest plane spanning trees. We focus on the specific problem of finding a longest plane (i.e. noncrossing) tree for a given set \mathcal{P} of $n \geq 3$ points in the plane in general position (i.e., no three points in \mathcal{P} are collinear). Such a tree is necessarily spanning. The general position assumption was also used in previous work [1, 16]; without it, one should specify whether overlapping edges are allowed, an additional complication that we would like to avoid.

If \mathcal{P} is in convex position, the longest plane tree for \mathcal{P} can be found in polynomial time on a real RAM, by adapting standard dynamic programming methods for plane structures on convex point sets [20, 22]. On the other hand, for an arbitrary point set \mathcal{P} , the problem is conjectured – but not known – to be NP-hard [1]. Hence, past research has focused on designing polynomial-time approximation algorithms. Typically, these algorithms construct several “simple” spanning trees for \mathcal{P} of small (unweighted) diameter, and one then argues that at least one such tree is sufficiently long. In a seminal work, Alon et al. [1] showed that a

longest star (a plane tree with diameter two) on \mathcal{P} yields a 0.5-approximation for the longest (not necessarily plane) spanning tree of \mathcal{P} . They further argued that this bound is essentially tight for point sets that consist of two large clusters far away from each other. Dumitrescu and Tóth [16] refined this algorithm by adding two additional families of candidate trees, now with diameter four. They showed that at least one member of this extended set of candidates provides a 0.502-approximation, which was further improved to 0.503 by Biniáz et al. [8]. In all these results, the approximation factor is analyzed by comparing the output of the algorithm with the length of a longest (not necessarily plane) spanning tree. Such a tree may be longer by a factor of up to $\pi/2 > 1.5$ than a maximum-length plane tree [1], as witnessed by, e.g., a large set of points spaced uniformly on a unit circle. While the ratio between the lengths of the longest plane tree and the longest (possibly crossing) tree is an interesting number in itself, the objective is to construct longest plane trees and thus it is better to compare the length of the constructed plane trees against the true optimum, that is, against the length of the longest plane tree. Considering certain trees of diameter at most five, a superset of the authors of this paper managed to compare against the longest plane tree and pushed the approximation factor to 0.512 [10]. This was subsequently improved even further to 0.519 by Biniáz [7].

Our results. We provide a deeper study of the interplay between the approximation factor and the diameter of the candidate trees. First, we give a polynomial-time algorithm to find a tree of diameter at most four that guarantees an approximation factor of roughly 0.546, a substantial improvement over the previous bounds.

► **Theorem 1.** *For any finite point set \mathcal{P} in general position (no three points collinear), we can compute in polynomial time a plane tree of Euclidean length at least $f \cdot |T_{\text{OPT}}|$, where $|T_{\text{OPT}}|$ denotes the length of a longest plane tree on \mathcal{P} and $f > 0.5467$ is the fourth smallest real root of the polynomial $P(x) = -80 + 128x + 504x^2 - 768x^3 - 845x^4 + 1096x^5 + 256x^6$.*

The algorithm “guesses” a longest edge of T_{OPT} and then constructs six trees: four stars and two more trees of diameter at most four. We show that one of these trees is always sufficiently long. The algorithm is very simple but its analysis uses several geometric insights. The polynomial $P(x)$ comes from optimizing the constants in the proof.

Second, we characterize longest plane trees for convex point sets. A *caterpillar* is a tree T that contains a path S , the *spine*, so that every vertex of $T \setminus S$ is adjacent to a vertex in S . A tree T that is straight-line embedded on a convex point set \mathcal{P} is a *zigzagging caterpillar* if its edges split the convex hull of \mathcal{P} into faces that are all triangles.

► **Theorem 2.** *If \mathcal{P} is convex then every longest plane tree on \mathcal{P} is a zigzagging caterpillar.*

► **Theorem 3.** *For any caterpillar C , there is a convex point set \mathcal{P} such that the unique longest tree for \mathcal{P} is isomorphic to C .*

In particular, the diameter of a (unique) longest plane tree is not bounded by any constant. As a consequence, we obtain an upper bound on the approximation factor $\text{BoundDiam}(d)$ that can be achieved by a plane tree of diameter at most d .

► **Theorem 4.** *For any $d \geq 2$, there is a convex point set \mathcal{P} so that every plane tree of diameter at most d on \mathcal{P} is at most*

$$\text{BoundDiam}(d) \leq 1 - \frac{6}{(d+1)(d+2)(2d+3)} = 1 - \Theta(1/d^3)$$

times as long as the length $|T_{\text{OPT}}|$ of a longest (unconstrained) plane tree on \mathcal{P} .

For small values of d , we have better bounds. For example, it is easy to see that $\text{BoundDiam}(2) \leq 1/2$: put two groups of roughly half of the points sufficiently far from each other. For $d = 3$, we can show $\text{BoundDiam}(3) \leq 5/6$.

► **Theorem 5.** *For any $\varepsilon > 0$, there is a convex point set \mathcal{P} such that every longest plane tree on \mathcal{P} of diameter 3 is at most $(5/6) + \varepsilon$ times as long as a longest (general) plane tree.*

Third, we give polynomial-time algorithms for finding a longest plane tree among those of diameter at most three and among a special class of trees of diameter at most four. Note that in contrast to diameter two, the number of spanning trees of diameter at most three is exponential in the number of points.

► **Theorem 6.** *For any set \mathcal{P} of n points in general position, a longest plane tree of diameter at most three on \mathcal{P} can be computed in $\mathcal{O}(n^4)$ time.*

► **Theorem 7.** *For any set \mathcal{P} of points in general position and any three specified points on the boundary of the convex hull of \mathcal{P} , we can compute in polynomial time a longest plane tree such that each edge is incident to at least one of the three specified points.*

The algorithms are based on dynamic programming. Even though the length $|T_{\text{OPT}}^3|$ of a longest plane tree of diameter at most three can be computed in polynomial time, we do not know the corresponding approximation factor $\text{BoundDiam}(3)$. The best bounds we are aware of are $1/2 \leq \text{BoundDiam}(3) \leq 5/6$. The lower bound follows from [1], the upper bound is from Theorem 5. We conjecture that $|T_{\text{OPT}}^3|$ actually gives a better approximation factor than the tree constructed in Theorem 1 – but we are unable to prove this.

Fourth, a natural way to design an algorithm for the longest plane spanning tree problem is the following local search heuristic [34]: start with an arbitrary plane tree T , and while it is possible, apply the following local improvement rule: if there are two edges e, f on \mathcal{P} such that $(T \setminus \{e\}) \cup \{f\}$ is a plane spanning tree for \mathcal{P} that is longer than T , replace e by f . Once no further local improvements are possible, output the current tree T . We show that for some point sets, this algorithm fails to compute the optimum answer as it may “get stuck” in a local optimum (see Lemma 17 in Section 5). This holds regardless of how the edges that are swapped are chosen. This suggests that a natural local search approach does not yield an optimal algorithm for the problem.

Preliminaries and notation. Let $\mathcal{P} \subset \mathbb{R}^2$ be a set of n points in the plane, so that no three points in \mathcal{P} are collinear. For any spanning tree T on \mathcal{P} , we denote by $|T|$ the total Euclidean edge length of T . Let T_{OPT} be a plane (i.e., noncrossing) spanning tree on \mathcal{P} with maximum Euclidean edge length. As the previous algorithms [1, 7, 8, 10, 16], we make extensive use of stars. The *star* S_p rooted at some point $p \in \mathcal{P}$ is the tree that connects p to all other points of \mathcal{P} .

We also need the notion of “flat” point sets. A point set \mathcal{P} is *flat* if $\text{diam}(\mathcal{P}) \geq 1$ and all y -coordinates in \mathcal{P} are essentially negligible, that is, their absolute values are bounded by an infinitesimal $\varepsilon > 0$. For flat point sets, we can approximate the length of an edge by subtracting the x -coordinates of its endpoints: the error becomes arbitrarily small as $\varepsilon \rightarrow 0$. Lastly, $D(p, r)$ denotes a closed disk with center p and radius r , while $\partial D(p, r)$ is its boundary: a circle of radius r centered at p .

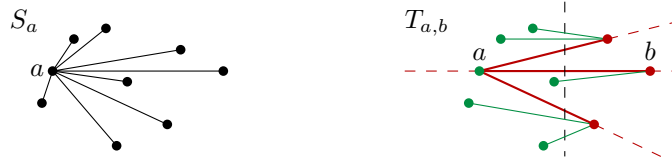


Figure 1 A tree S_a and a tree $T_{a,b}$.

2 An improved approximation algorithm

We present a polynomial-time algorithm that yields an $f \doteq 0.5467$ -approximation of a longest plane tree for general point sets and a $(2/3)$ -approximation for flat point sets. We consider the following trees $T_{a,b}$, for $a, b \in \mathcal{P}$ (see Figure 1): let \mathcal{P}_a be the points of \mathcal{P} closer to a than to b , and let $\mathcal{P}_b = \mathcal{P} \setminus \mathcal{P}_a$. First, connect a to every point in \mathcal{P}_b . Then, connect each point of $\mathcal{P}_a \setminus \{a\}$ to some point of \mathcal{P}_b without introducing crossings. This yields a tree of diameter at most four. In general, $T_{a,b}$ and $T_{b,a}$ are different and neither is uniquely determined, but for $\mathcal{P}_a = \{a\}$ both $T_{a,b}$ and $T_{b,a}$ coincide with the star S_a .

Our algorithm $\text{AlgSimple}(\mathcal{P})$ computes all stars S_a and the tree $T_{a,b}$, for each ordered pair $a, b \in \mathcal{P}$, and it returns a longest one. The algorithm runs in polynomial time, as there are $\mathcal{O}(n^2)$ relevant trees, each of which can be found in polynomial time.

Given multiple trees that all contain a common edge ab , we direct all other edges towards ab and assign to each point in $\mathcal{P} \setminus \{a, b\}$ its unique outgoing edge. The edge ab remains undirected. Denote the length of the edge assigned to $p \in \mathcal{P} \setminus \{a, b\}$ in such a tree T by $\ell_T(p)$.

Theorem 1 states that for any \mathcal{P} , we have $|T_{\text{ALG}}| > 0.5467 \cdot |T_{\text{OPT}}|$. As a warm-up for the full proof, we first show a stronger result for the special case of flat point sets: if \mathcal{P} is flat, we have $|T_{\text{ALG}}| \geq (2/3) \cdot |T_{\text{cr}}|$, where T_{cr} is a longest (possibly crossing) tree. In fact, the constant $2/3$ is tight when comparing to T_{cr} :

► **Observation 8.** *There is an infinite family of point sets $\mathcal{P}_1, \mathcal{P}_2, \dots$ with $|\mathcal{P}_n| = 2n$ and*

$$\lim_{n \rightarrow \infty} \frac{|T_{\text{OPT}}|}{|T_{\text{cr}}|} \leq \frac{2}{3}.$$

Proof. Let $\mathcal{P}_n = \{p_1, \dots, p_{2n}\}$ be a flat point set where the points p_i are spaced evenly on a convex arc with x -coordinates $1, \dots, 2n$, see Figure 2. It can be shown inductively, that the star S_{p_1} is a longest plane spanning tree and thus $|T_{\text{OPT}}| = |S_{p_1}| = \sum_{i=1}^{2n-1} i = 2n^2 - n$. On the other hand, the right side in Figure 2 shows a crossing spanning tree of total length $(2n - 1) + 2 \sum_{i=n}^{2n-2} i = 3n^2 - 3n + 1 \leq |T_{\text{cr}}|$. ◀

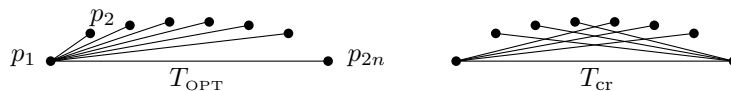
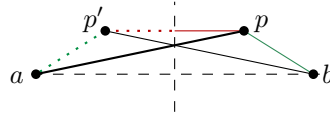


Figure 2 The point set \mathcal{P}_n of $2n$ points with equally spaced x -coordinates $1, 2, \dots, 2n$, with a longest plane and the longest general spanning tree.

► **Theorem 9.** *Suppose \mathcal{P} is flat. Then,*

$$|T_{\text{ALG}}| \geq \frac{2}{3} |T_{\text{cr}}| \geq \frac{2}{3} |T_{\text{OPT}}|.$$



■ **Figure 3** By triangle inequality and symmetry, we have $\|pp'\| + \|pb\| \geq \|p'b\| = \|pa\|$.

Proof. As $|T_{\text{cr}}| \geq |T_{\text{OPT}}|$, it suffices to show the first inequality. Denote the diameter of \mathcal{P} by ab (see Figure 3). Consider the four trees $S_a, T_{a,b}, T_{b,a}, S_b$. It suffices to show that there exists a $\beta \in (0, 1/2)$ such that

$$(1/2 - \beta)|S_a| + \beta|T_{a,b}| + \beta|T_{b,a}| + (1/2 - \beta)|S_b| \geq \frac{2}{3} \cdot |T_{\text{cr}}|.$$

Here we fix $\beta = \frac{1}{3}$ and equivalently show:

$$\frac{|S_a| + 2|T_{a,b}| + 2|T_{b,a}| + |S_b|}{6} \geq \frac{2}{3} \cdot |T_{\text{cr}}| \quad (1)$$

which is enough, as

$$\max\{|S_a|, |T_{a,b}|, |T_{b,a}|, |S_b|\} \geq \frac{1}{6}(|S_a| + 2|T_{a,b}| + 2|T_{b,a}| + |S_b|)$$

The trees $S_a, T_{a,b}, T_{b,a}, S_b$ all contain the edge ab , and since that edge realizes the diameter, we can assume that T_{cr} also contains ab . We fix a $p \in \mathcal{P} \setminus \{a, b\}$, assume without loss of generality that $\|pa\| \geq \|pb\|$, and denote by p' the reflection of p at the perpendicular bisector of ab (see Figure 3). Using the notation $\ell_T(p)$ from above,

$$\begin{aligned} \frac{1}{6}(\ell_{S_a}(p) + 2\ell_{T_{a,b}}(p) + 2\ell_{T_{b,a}}(p) + \ell_{S_b}(p)) &\geq \frac{1}{6}(\|pa\| + 2\|pa\| + \|pp'\| + \|pb\|) \\ &\geq \frac{1}{6}(3\|pa\| + \|p'b\|) = \frac{2}{3} \cdot \|pa\| \geq \frac{2}{3} \cdot \ell_{T_{\text{cr}}}(p). \end{aligned}$$

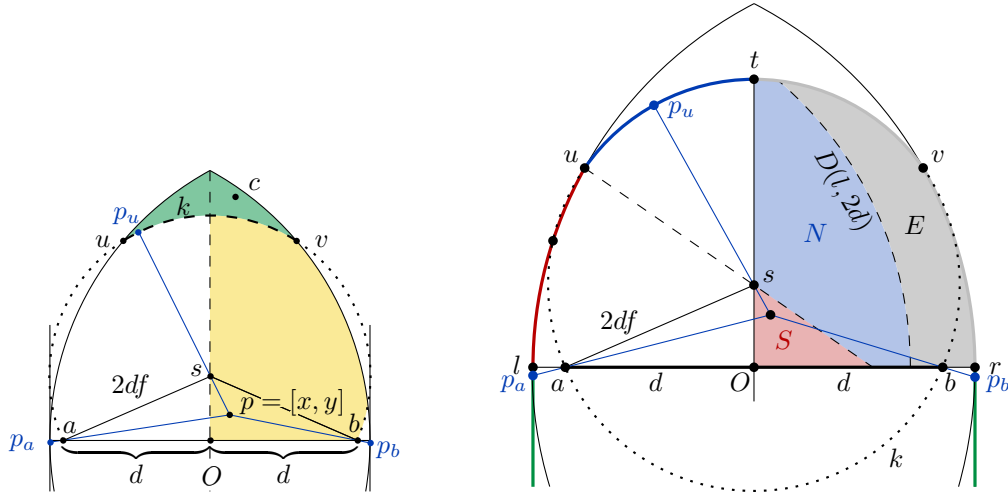
Here, we used in the first step that $\ell_{S_a}(p) = \ell_{T_{a,b}}(p) = \|pa\|$, $\ell_{T_{b,a}}(p) \geq \|p'p\|/2$, and $\ell_{S_b}(p) = \|pb\|$. In the second and third step, we used the triangle inequality $\|pp'\| + \|pb\| \geq \|p'b\|$ and the symmetry $\|p'b\| = \|pa\|$. The final step follows since \mathcal{P} is flat and hence $\ell_{T_{\text{cr}}}(p) \leq \max\{\|pa\|, \|pb\|\} = \|pa\|$. Now, (1) follows by summing over all $p \in \mathcal{P} \setminus \{a, b\}$. ◀

► **Theorem 1.** *For any finite point set \mathcal{P} in general position (no three points collinear), we can compute in polynomial time a plane tree of Euclidean length at least $f \cdot |T_{\text{OPT}}|$, where $|T_{\text{OPT}}|$ denotes the length of a longest plane tree on \mathcal{P} and $f > 0.5467$ is the fourth smallest real root of the polynomial $P(x) = -80 + 128x + 504x^2 - 768x^3 - 845x^4 + 1096x^5 + 256x^6$.*

Proof. We outline the proof strategy, referring to lemmas that will formally be stated later in this section. Without loss of generality, suppose \mathcal{P} has diameter 2. Consider a longest edge ab of T_{OPT} and denote its length by $2d$ (we have $d \leq 1$).

Let $u, v \in \mathcal{P}$ be two points realizing the diameter of \mathcal{P} . Note that in general the longest edge of T_{OPT} does not realize the diameter and thus a, b and u, v differ. If $2df \leq 1$, it follows from previous work that one of S_u or S_v is long enough (see [10, Lemma 2.1]). Thus, we henceforth assume that $2df > 1$. Note that \mathcal{P} lies in the lens $L = D(a, 2) \cap D(b, 2)$ and that the points a and b are in L . Choose a coordinate system with $a = (-d, 0)$ and $b = (d, 0)$, and let s, s' be the two points on the y -axis with $\|sa\| = \|sb\| = \|s'a\| = \|s'b\| = 2df$, where s is

the point above the x -axis. Since $2df > 1$, the circles $k = \partial D(s, 2df)$ and $k' = \partial D(s', 2df)$ intersect the boundary of L . Let u, v and u', v' be the intersection points above and below the x -axis respectively, so that u and u' are to the left of the y -axis. The *far region* consists of the points in L above the arc of k between u and v in clockwise direction and of the points in L below the arc of k' between u' and v' in counter-clockwise direction. The *truncated lens* contains the remaining points, see Figure 4a.



(a) The lens is split into the far region (green) and the truncated lens. (b) The truncated lens is further subdivided into three regions E, N and S .

■ **Figure 4** Subdivision of the lens.

In Lemma 10, we argue that if the far region contains a point $c \in \mathcal{P}$, then one of the three stars S_a, S_b , or S_c is long enough. Otherwise, if all of \mathcal{P} lies in the truncated lens, we claim that one of the trees $S_a, T_{a,b}, T_{b,a}$, or S_b is long enough. These four trees all contain the edge ab . Thus, we can again use the notation $\ell_T(p)$ from above to define for any $p \in \mathcal{P} \setminus \{a, b\}$ and for any $\beta \in (0, 1/2)$, the weighted average

$$\text{avg}(p, \beta) = (1/2 - \beta) \cdot \ell_{S_a}(p) + \beta \cdot \ell_{T_{a,b}}(p) + \beta \cdot \ell_{T_{b,a}}(p) + (1/2 - \beta) \cdot \ell_{S_b}(p).$$

To finish the argument, we aim to find a $\beta \in (0, 1/2)$ so that for any $p \in \mathcal{P} \setminus \{a, b\}$, we have $\text{avg}(p, \beta) \geq f \cdot \ell_{T_{\text{OPT}}}(p)$ (note that $\ell_{T_{\text{OPT}}}(p)$ is defined, since ab is an edge of T_{OPT}). In contrast to the proof for Theorem 9, this now requires much more work. After that, the approximation guarantee follows by considering the sum $\sum_{p \in \mathcal{P} \setminus \{a, b\}} \text{avg}(p, \beta)$, as before.

For proving $\text{avg}(p, \beta) \geq f \cdot \ell_{T_{\text{OPT}}}(p)$, we can without loss of generality assume that $p = (x, y)$, with $x, y \geq 0$. The following definitions are illustrated in Figure 4a. Let p_a be the point with x -coordinate $-(2 - d)$ on the ray pa . If $x < d$, let p_b be the point with x -coordinate $2 - d$ on the ray pb . Otherwise, the ray pb does not intersect the vertical line with x -coordinate $2 - d$, and we set $p_b = b$. Additionally, define p_u to be the furthest point from p on the portion of the boundary of the far region that is contained in the circle $k = \partial D(s, 2df)$. The proof now proceeds in the following steps:

1. we show that $\ell_{T_{\text{OPT}}}(p) \leq \min \{2d, \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}\}$ (Lemma 11);
2. we show that the term $\|pp_b\|$ in this upper bound can be omitted (Lemma 12);
3. we establish a lower bound on $\text{avg}(p, \beta)$ (Lemma 13); and
4. we use this lower bound to find constraints on β that ensure $\text{avg}(p, \beta) \geq f \cdot \min\{2d, \|pp_a\|\}$ and $\text{avg}(p, \beta) \geq f \cdot \min\{2d, \|pp_u\|\}$, respectively (Lemmas 14 and 15).

It then remains to show that there exists a β that satisfies both the constraints from Lemma 14 and from Lemma 15. It turns out that this holds for any $\beta \in (0, 1/2)$ with

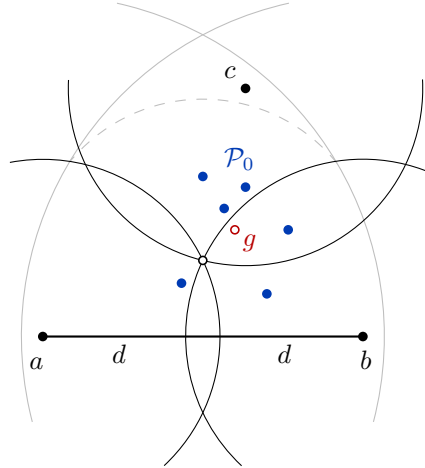
$$(2f - 1) / (2\sqrt{5 - 8f} - 1) \leq \beta \leq 1 - f\sqrt{4f^2 - 1} - 2f^2. \tag{2}$$

Our choice of f ensures that the two expressions in (2) have the same value (≈ 0.1604). Setting β accordingly, we get the desired approximation (cf. the full version for the calculation). ◀

It remains to prove Lemmas 10 to 15. Their statements rely on the notation introduced in the proof outline of Theorem 1, so we recommend to first consult the paragraphs above.

► **Lemma 10.** *Let ab (with $\|ab\| = 2d$) be the longest edge of T_{OPT} . If \mathcal{P} contains a point c in the far region, then $\max\{|S_a|, |S_b|, |S_c|\} \geq f \cdot |T_{\text{OPT}}|$.*

Proof. By the definition of the far region, the triangle abc is acute-angled and its circumradius R satisfies $R \geq 2df$. Let $g = \frac{1}{|\mathcal{P}_0|} \sum_{p \in \mathcal{P}_0} p$ be the center of mass of the point set $\mathcal{P}_0 \equiv \mathcal{P} \setminus \{a, b, c\}$, see Figure 5. Since the triangle abc is acute-angled, it has a vertex v with $\|vg\| \geq R$. By definition of g , we have $\sum_{p \in \mathcal{P}_0} \vec{vp} = |\mathcal{P}_0| \cdot \vec{vg}$, and the triangle inequality



■ **Figure 5** Lemma 10. In the illustration, \mathcal{P}_0 consists of 6 points and we can take $v = a$. The common point of the three black circles is the circumcenter of triangle abc .

gives $\sum_{p \in \mathcal{P}_0} \|vp\| \geq |\mathcal{P}_0| \cdot \|vg\| \geq (n - 3) \cdot R$. As $\|va\| + \|vb\| + \|vc\| \geq 2R$ holds in any acute-angled triangle, we obtain $|S_v| \geq (n - 1) \cdot R \geq (n - 1) \cdot 2df \geq f \cdot |T_{\text{OPT}}|$. ◀

► **Lemma 11.** *For every point $p = (x, y)$ with $x, y \geq 0$ in the truncated lens, we have $\ell_{T_{\text{OPT}}}(p) \leq \min\{2d, \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}\}$.*

Proof Sketch. (Full proof in the full version) Let $l = (d - 2, 0)$ and $r = (2 - d, 0)$ be the left- and rightmost points of $D(a, 2) \cap D(b, 2)$. We divide the truncated lens into further regions (see Figure 4b): the region E lies inside the truncated lens but outside of $D(l, 2d)$, and the remainder of the truncated lens is divided into the part N above the line us and the part S below us . If $p \in E$, then $\min\{2d, \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}\} = 2d$, and we are done, since $\ell_{T_{\text{OPT}}}(p) \leq 2d$. Next, assume that $p \in N \cup S$, and let p_f be the furthest point from p in the truncated lens. An exhaustive case distinction over the quadrant containing p_f shows that $\|pp_f\| \leq \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}$, which proves the lemma. ◀

This bound can be simplified by using the following lemma:

► **Lemma 12.** *For every point $p = (x, y)$ with $x, y \geq 0$ in the truncated lens, if $\|pp_a\| \leq 2d$, then $\|pp_b\| \leq \|pp_a\|$.*

The algebraic proof can be found in the full version.

Now we give a general lower bound on $\text{avg}(p, \beta)$ that we will use in Lemmas 14 and 15.

► **Lemma 13.** *Let $p = (x, y) \in \mathbb{R}^2$ be a point with $x, y \geq 0$, and let $\beta \in (0, 1/2)$. Then,*

$$\text{avg}(p, \beta) \geq \frac{d \cdot (1 - \beta) + x \cdot 2\beta}{d + x} \cdot \|pa\|.$$

Proof Sketch. (Full proof in the full version) We expand the definition and replace the $\ell_T(p)$ -terms by $\|pa\|$, $\|pb\|$, and x , respectively. By similar geometric arguments as in Theorem 9,

$$\text{avg}(p, \beta) \geq (1/2) \cdot \|pa\| + (\beta/2) \cdot \|pa\| + ((1/2) - (3/2)\beta) \cdot \|pb\|.$$

Using $\|pb\| \geq \frac{d-x}{d+x} \cdot \|pa\|$, we get the desired

$$\text{avg}(p, \beta) \geq \frac{(1 + \beta)(d + x) + (1 - 3\beta)(d - x)}{2(d + x)} \cdot \|pa\| = \frac{(1 - \beta) \cdot d + 2\beta \cdot x}{d + x} \cdot \|pa\|. \quad \blacktriangleleft$$

► **Lemma 14.** *Let $p = (x, y)$ be any point in the truncated lens with $x, y \geq 0$. Then, if $\frac{2f-1}{5-8f} \leq \beta \leq \frac{1}{2} \cdot f$, we have $\text{avg}(p, \beta) \geq f \cdot \min\{2d, \|pp_a\|\}$.*

Proof Sketch. (Full proof in the full version) We show that if $x \geq 3d - 2$, then $\text{avg}(p, \beta) \geq f \cdot 2d$, and if $x \leq 3d - 2$, then $\text{avg}(p, \beta) \geq f \cdot \|pp_a\|$. Using Lemma 13, both cases reduce to the following inequality, which holds by the assumption on β :

$$\beta \cdot (5d - 4) \geq \beta \cdot (5d - 8df) \geq \frac{2f - 1}{5 - 8f} \cdot d \cdot (5 - 8f) = d(2f - 1). \quad \blacktriangleleft$$

► **Lemma 15.** *Let $p = (x, y)$ be any point in the truncated lens with $x, y \geq 0$. Suppose that $\beta < \frac{151}{304} \cdot f$ and that $\frac{1}{2} \leq f \leq \frac{19}{32}$, then $\text{avg}(p, \beta) \geq f \cdot \min\{2d, \|pp_u\|\}$, if*

$$\frac{2f - 1}{2\sqrt{5 - 8f} - 1} \leq \beta \leq 1 - f\sqrt{4f^2 - 1} - 2f^2.$$

Proof Sketch. (Full proof in the full version) By Lemma 13, it suffices to show that

$$\lambda = \frac{d \cdot (1 - \beta) + x \cdot 2\beta}{d + x} \cdot \|pa\| \geq f \cdot \min\{2d, \|pp_u\|\}. \quad (3)$$

Case 1: $y \leq y(u)$. λ is an increasing function in y and $\|pp_u\|$ is a decreasing function in y , for $y \leq y(u)$. Thus, it suffices to show (3) for $y = 0$. In this case, λ becomes $\lambda_0 = d \cdot (1 - \beta) + x \cdot 2\beta$, which is positive. Let $q = (q_x, 0)$, $q_x \geq 0$, be the point on the x -axis with $\|qs\| = 2d(1 - f)$. For $p = q$, we have $\|pp_u\| \leq \|ps\| + \|sp_u\| = 2d$.

Case 1a: $0 \leq x \leq q_x$. The Pythagorean theorem and the bounds on β yield $\lambda_0 \geq f \cdot \|pp_u\|$.

Case 1b: $q_x < x$. It suffices to show $\lambda_0 \geq f \cdot 2d$, for $x = q_x$. This follows from Case 1a.

Case 2: $y > y(u)$ Now, $\|pp_u\| = \|pu\| \leq \|uv\|$. Also, we have $x(u) \geq -d$ and $x \leq d$, which gives $\min\{2d, \|pp_u\|\} = \|pp_u\|$. Thus, (3) becomes $\lambda \geq f \cdot \|pp_u\|$. From $y > y(u)$, we get $\|pa\| \geq \|pp_u\|$, so we need $\lambda/\|pa\| \geq f$. This follows by straightforward algebra. \blacktriangleleft

3 Convex and flat convex point sets

We present two results for convex point sets: (i) if \mathcal{P} is convex, any longest plane tree is a caterpillar, and any caterpillar appears as the unique longest plane tree of a convex point set; and (ii) by looking at suitable flat convex sets, we prove upper bounds on the approximation factor achieved by the longest plane tree among those with diameter at most d .

Convex sets and caterpillars. A tree C is called *caterpillar* if it contains a path P such that every node in $C \setminus P$ is adjacent to a node on P . We consider trees that span a given convex point set \mathcal{P} . We call (a drawing of) such a tree T a *zigzagging caterpillar* if T is a caterpillar and the *dual graph* T^* of T is a path, where T^* is defined as follows: consider a smooth closed curve through all points of \mathcal{P} . The curve bounds a convex region that is split by the $n - 1$ edges of T into n subregions. Then T^* has a node for each such subregion and two nodes are connected if their subregions share an edge of T (see Figure 6).

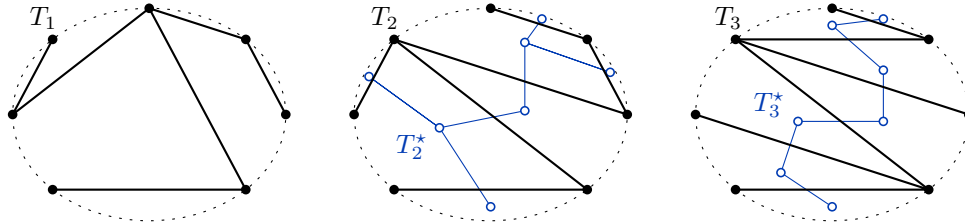


Figure 6 T_1 is spanning \mathcal{P} but it is not a caterpillar. T_2 is a caterpillar but it is not zigzagging. T_3 is a zigzagging caterpillar, since the dual tree T_3^* is a path.

► **Theorem 2.** *If \mathcal{P} is convex then every longest plane tree on \mathcal{P} is a zigzagging caterpillar.*

Proof. Let T_{OPT} be a longest plane tree. We prove that T_{OPT}^* is a path. Suppose not, and consider a node in T_{OPT}^* of degree at least 3. Let ab, bc, cd be three corresponding edges of T_{OPT} . As $abcd$ is a convex quadrilateral, the triangle inequality gives $\|ab\| + \|cd\| < \|ac\| + \|bd\|$, so $\|ab\| < \|ac\|$ or $\|cd\| < \|bd\|$ (or both). Now, $T_1 = T_{\text{OPT}} \cup ac \setminus ab$ and $T_2 = T_{\text{OPT}} \cup bd \setminus cd$ are plane trees, and at least one of them is longer than T_{OPT} , a contradiction. ◀

Note that as \mathcal{P} is assumed to be convex in this context, an optimal caterpillar can be found by applying the dynamic programming approach for the convex case described in Section 4.

Conversely, for every caterpillar C we construct a convex set \mathcal{P}_C whose longest plane tree is isomorphic to C . In fact, \mathcal{P}_C will be a *flat arc*: a flat convex point set $\{a_i = (x_i, y_i)\}_{i=1}^{m+1}$, where $x_i < x_j$, for $i < j$. The sequence $G(\mathcal{P}_C) = \{g_i\}_{i=1}^m = \{|x_{i+1} - x_i|\}_{i=1}^m$ is the *gap sequence* of \mathcal{P}_C . Given a spanning tree T for \mathcal{P}_C , we define its *cover sequence* $\text{Cov}(T) = \{c_i\}_{i=1}^m$ where c_i denotes the number of times gap g_i is “covered”, see Figure 7. Then, $|T| = \sum_{i=1}^m c_i \cdot g_i$.

► **Lemma 16.** *Consider a flat arc $\{a_1, \dots, a_{m+1}\}$ and a zigzagging caterpillar T containing the edge a_1a_{m+1} . Then the sequence $\text{Cov}(T)$ is a unimodal permutation of $\{1, 2, \dots, m\}$.*

Proof. We show this lemma by induction on m . The case $m = 1$ is clear. Fix $m \geq 2$. By the definition of a zigzagging caterpillar, the dual graph T^* of T is a path. Since, by the assumption of the lemma, a_1a_{m+1} is an edge of T , either a_1a_m or a_2a_{m+1} is an edge of T too. Without loss of generality assume a_1a_m is an edge of T . Then $T \setminus \{a_1a_{m+1}\}$ is a zigzagging caterpillar on m points a_1, \dots, a_m containing the edge a_1a_m , hence by induction its cover

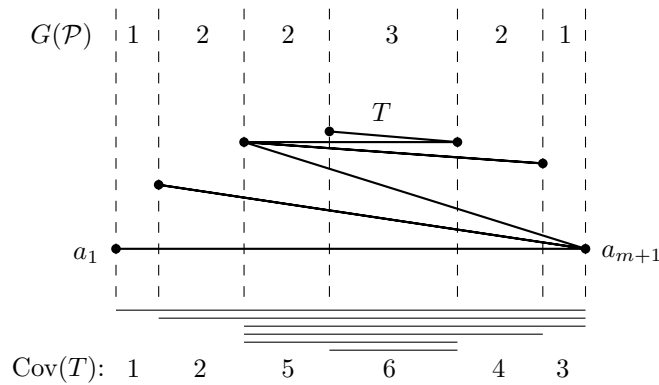


Figure 7 A tree with its gap and cover sequence.

sequence is a unimodal permutation of $\{1, 2, \dots, m - 1\}$. Adding the omitted edge $a_1 a_{m+1}$ adds 1 to each of the $m - 1$ elements and appends a 1 to the list, giving rise to a unimodal permutation of $\{1, 2, \dots, m\}$. This completes the proof. ◀

► **Theorem 3.** *For any caterpillar C , there is a convex point set \mathcal{P} such that the unique longest tree for \mathcal{P} is isomorphic to C .*

Proof. Consider a flat arc $\mathcal{P} = \{a_1, \dots, a_{m+1}\}$, with a yet unspecified gap sequence $\{g_i\}_{i=1}^m$, and let T be a drawing of C onto \mathcal{P} that contains the edge $a_1 a_{m+1}$ and is zigzagging (such a drawing always exists). By Lemma 16, the cover sequence $\text{Cov}(T) = \{c_i\}_{i=1}^m$ is a unimodal permutation of $\{1, 2, \dots, m\}$. The total length of T can be expressed as $|T| = \sum_{i=1}^m c_i \cdot g_i$.

Now we specify the gap sequence: for $i = 1, \dots, m$, set $g_i = c_i$. It remains to show that T constitutes the longest plane tree T_{OPT} of \mathcal{P} .

By Theorem 2, T_{OPT} is a zigzagging caterpillar. Also, $a_1 a_{m+1}$ is an edge of T_{OPT} : suppose not. Since $a_1 a_{m+1}$ does not cross any other edge, adding it to T_{OPT} produces a plane graph with a single cycle C . All edges of T_{OPT} are shorter than $a_1 a_{m+1}$, so omitting any other edge from C yields a longer plane tree, a contradiction. We can thus apply Lemma 16 to see that $\text{Cov}(T_{\text{OPT}})$ is a unimodal permutation π of $\{1, 2, \dots, m\}$ and that $|T_{\text{OPT}}| = \sum_{i=1}^m \pi_i \cdot g_i$. As c_i and g_i match and as c, g , and π are permutations, the Cauchy-Schwarz inequality gives

$$|T_{\text{OPT}}| = \sum_{i=1}^m \pi_i \cdot g_i \leq \sqrt{\sum_{i=1}^m \pi_i^2 \cdot \sum_{i=1}^m g_i^2} = \sum_{i=1}^m c_i^2 = |T|, \text{ with equality iff } \pi_i = c_i, \text{ for all } i. \quad (4)$$

Therefore T_{OPT} is unique and $T_{\text{OPT}} = T$ as desired. ◀

Upper bounds on BoundDiam(d). The algorithms for approximating $|T_{\text{OPT}}|$ often produce trees with small diameter. Given $d \geq 2$ and a point set \mathcal{P} , let $T_{\text{OPT}}^d(\mathcal{P})$ be a longest plane tree on \mathcal{P} among those with diameter at most d . We ask what is the approximation ratio

$$\text{BoundDiam}(d) = \inf_{\mathcal{P}} \frac{|T_{\text{OPT}}^d(\mathcal{P})|}{|T_{\text{OPT}}(\mathcal{P})|}.$$

For $d = 2$, this question concerns the performance of stars. A result of Alon, Rajagopalan, and Suri [1, Theorem 4.1] can be restated as $\text{BoundDiam}(2) = 1/2$. We show a crude upper bound on $\text{BoundDiam}(d)$ for general d and a specific upper bound for the case $d = 3$. (Note that Theorem 6 shows that for any fixed \mathcal{P} we can compute $|T_{\text{OPT}}^3(\mathcal{P})|$ in polynomial time.) Our proofs use the notions of flat arc, gap sequence, and cover sequence defined above.

► **Theorem 4.** For any $d \geq 2$, there is a convex point set \mathcal{P} so that every plane tree of diameter at most d on \mathcal{P} is at most

$$\text{BoundDiam}(d) \leq 1 - \frac{6}{(d+1)(d+2)(2d+3)} = 1 - \Theta(1/d^3)$$

times as long as the length $|T_{\text{OPT}}|$ of a longest (unconstrained) plane tree on \mathcal{P} .

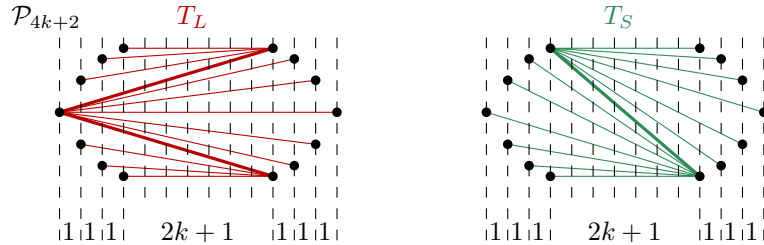
Proof. Let \mathcal{P} be a flat arc on $d+2$ points with gap sequence $G = (1, 3, 5, \dots, d+1, \dots, 6, 4, 2)$. Since G is unimodal, we can argue as in the proof of Theorem 3 to see that T_{OPT} is the zigzagging caterpillar whose cover sequence is G , i.e., a path with $d+1$ edges (and diameter $d+1$). Moreover, this path is the only optimal plane tree spanning the flat arc because of Theorem 2 and the Cauchy-Schwarz inequality; see the argument leading to (4). Therefore, any other plane spanning tree $T \neq T_{\text{OPT}}$, zigzagging caterpillar or not, has an integer length less than $|T_{\text{OPT}}|$. Using $|T_{\text{OPT}}| = \sum_{i=1}^{d+1} i^2 = \frac{1}{6}(d+1)(d+2)(2d+3) = \frac{1}{3}d^3 + o(d^3)$, we obtain

$$\text{BoundDiam}(d) \leq \frac{|T_{\text{OPT}}| - 1}{|T_{\text{OPT}}|} = 1 - \frac{6}{(d+1)(d+2)(2d+3)} = 1 - \Theta(1/d^3). \quad \blacktriangleleft$$

For $d = 3$, Theorem 4 gives $\text{BoundDiam}(3) \leq 29/30$. By tailoring the point set size, the gap sequence $\{g_i\}_{i=1}^m$, and by considering non-arcs, we improve this to $\text{BoundDiam}(3) \leq 5/6$.

► **Theorem 5.** For any $\varepsilon > 0$, there is a convex point set \mathcal{P} such that every longest plane tree on \mathcal{P} of diameter 3 is at most $(5/6) + \varepsilon$ times as long as a longest (general) plane tree.

Proof (Sketch). (Full proof in the full version) Let \mathcal{P}_{4k+2} consist of two flat arcs, symmetric with respect to a horizontal line, each with a gap sequence $\underbrace{1, \dots, 1}_{k \times}, 2k+1, \underbrace{1, \dots, 1}_{k \times}$. In other words, \mathcal{P}_{4k+2} consists of two diametrically opposing points, four unit-spaced arcs of k points each, and a large central gap of length $2k+1$ (see Figure 8).

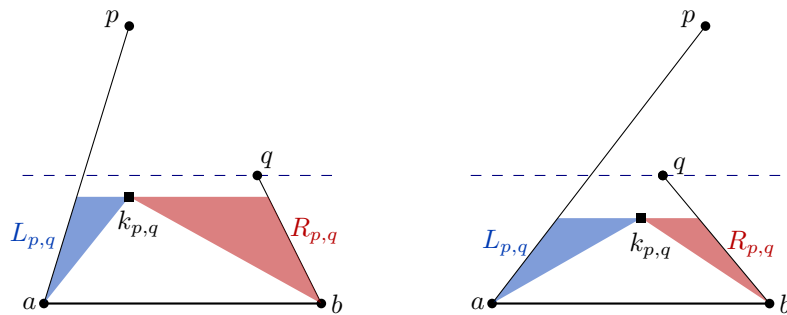


■ **Figure 8** An illustration of the point set \mathcal{P}_{4k+2} when $k = 3$, with trees T_L (red) and T_S (green).

On the one hand, straightforward counting gives $|T_{\text{OPT}}| \geq |T_L| = 12k^2 + 6k + 1$, where T_L is the tree depicted in Figure 8. On the other hand, any tree with diameter at most 3 is either a star or it contains an edge ab such that each other point of \mathcal{P} is connected either to a or to b . For a star T , simple computation gives $|T| \leq 8k^2 + 6k + 1$. For the other case, one can show that the longest tree is obtained when points a, b lie on the opposite sides of the large central gap and at least one of them lies on the boundary of this gap, as is the case for instance for the tree T_S depicted in Figure 8. We have $|T_S| = 10k^2 + 6k + 1$, thus

$$\text{BoundDiam}(3) \leq \frac{10k^2 + 6k + 1}{12k^2 + 6k + 1},$$

which tends to $5/6$ as $k \rightarrow \infty$. ◀



■ **Figure 9** Fixing $k_{p,q}$ gives two possible triangular regions where edges are forced.

4 Polynomial time algorithms for small diameter

We show how to compute a longest tree of diameter at most three in polynomial time, using dynamic programming. The main challenge is to devise an appropriate partition into independent subproblems. Our approach bears some resemblance to the polynomial time plane matching algorithm of Aloupis et al. [2]. The main challenge in our case is the efficient implementation of the dynamic program.

Our approach extends to a certain class of diameter-four trees, see the full version of this paper. Every tree of diameter two or three is a *bistar*, that is, it contains two vertices a and b so that every edge is incident to at least one of a or b . To prove Theorem 6, we note that there are $\Theta(n^2)$ choices for a and b , and we show how to compute a longest bistar rooted at a fixed pair a, b in $\mathcal{O}(n^2)$ time.

Without loss of generality, we can assume that the points a and b lie on a horizontal line with a to the left of b . As no edge will cross this line, we can also assume that all points lie above this line.

The subproblems for the dynamic program are indexed by ordered pairs p, q of distinct points from \mathcal{P} , so that the line segments ap and bq do not cross. A pair that satisfies this condition is a *valid pair*. For each valid pair p, q , the segments ap , pq , qb , and ba form a simple (possibly non-convex) quadrilateral. Let $Q(p, q)$ be the (convex) portion of this quadrilateral below the horizontal line $y = \min\{y(p), y(q)\}$. We define the value $Z(p, q)$ as the length of the longest plane bistar rooted at a and b on the points in the interior of $Q(p, q)$, without counting $\|ab\|$. If there are no points of \mathcal{P} within the quadrilateral $Q(p, q)$, we set $Z(p, q) = 0$.

If the quadrilateral $Q(p, q)$ contains some points from \mathcal{P} , we let $k_{p,q}$ be the highest point of \mathcal{P} inside of $Q(p, q)$. If we connect $k_{p,q}$ to a , we force all points in the triangle $L_{p,q}$ defined by the edges ap and $ak_{p,q}$ and the line $y = y(k_{p,q})$ to be connected to a . Similarly, when connecting $k_{p,q}$ to b , we force the triangle $R_{p,q}$ defined by bq , $bk_{p,q}$ and the line $y = y(k_{p,q})$; see Figure 9. In the former case, we are left with the subproblem defined by the valid pair $k_{p,q}, q$, while in the latter case we are left with the subproblem defined by the valid pair $p, k_{p,q}$. This yields the following recurrence for each valid pair p, q :

$$Z(p, q) = \begin{cases} 0, & \text{if no point of } \mathcal{P} \text{ is in } Q(p, q), \\ \max \begin{cases} Z(k_{p,q}, q) + \|ak_{p,q}\| + \sum_{l \in L_{p,q}} \|al\| \\ Z(p, k_{p,q}) + \|bk_{p,q}\| + \sum_{r \in R_{p,q}} \|br\| \end{cases}, & \text{otherwise.} \end{cases}$$

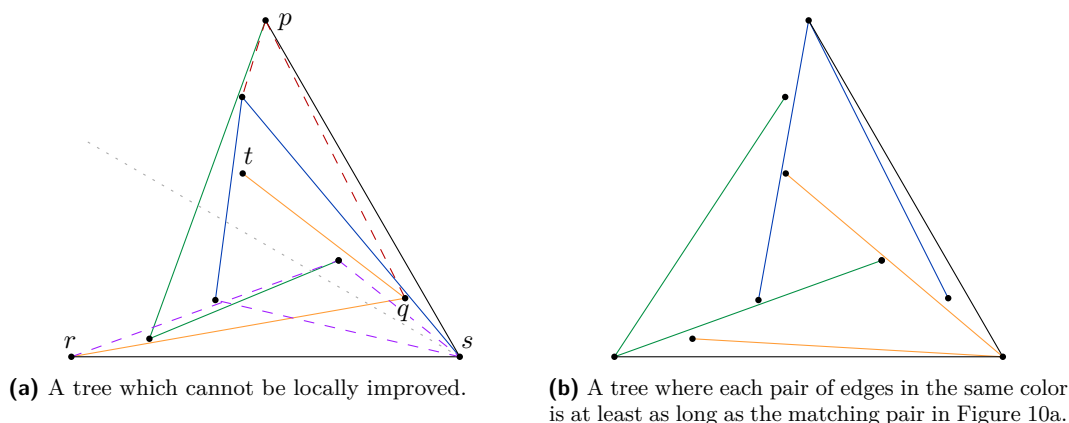
Using the values $Z(p, q)$, for all valid p, q , together with a specialized approach to solve the relevant range searching problems, we can show that a longest plane bistar for a fixed pair a, b of vertices can be computed in $\mathcal{O}(n^2)$ time; for details see the full version.

5 Local improvements fail

One could hope that the longest plane spanning tree problem could perhaps be solved by either a greedy approach or by a local search approach [34]. It is easy to find point sets on as few as 5 points where the obvious greedy algorithm fails to find the longest plane tree. In this section, we show that the following natural local search algorithm $\text{AlgLocal}(\mathcal{P})$ fails too:

■ **Algorithm 1** $\text{AlgLocal}(\mathcal{P})$.

-
1. Construct an arbitrary plane spanning tree T on \mathcal{P} .
 2. **While** there exists a pair of points a, b such that $T \cup \{ab\}$ contains an edge cd with $|cd| < |ab|$ and $T \cup \{ab\} \setminus \{cd\}$ is a plane spanning tree:
 - a. Set $T \rightarrow T \cup \{ab\} \setminus \{cd\}$. // tree $T \cup \{ab\} \setminus \{cd\}$ is longer than T
 3. Output T .
-



■ **Figure 10** The algorithm $\text{AlgLocal}(\mathcal{P})$ can get stuck.

► **Lemma 17.** *There are point sets \mathcal{P} for which the algorithm $\text{AlgLocal}(\mathcal{P})$ fails to compute the longest plane tree.*

Proof. We construct a point set \mathcal{P} consisting of 9 points to show the claim. The points are placed on three concentric equilateral which are slightly rotated, see Figure 10.

Now consider the tree on this point set depicted by the solid edges in Figure 10a. Note that the green, blue and yellow edges are rotational symmetric. A simple case distinction, using the dashed edges as prototype for different non-edges shows that $\text{AlgLocal}(\mathcal{P})$ stops at this tree. On the other hand, in the tree depicted in Figure 10b each pair of the same colored edges is longer than its counterpart in Figure 10a. Therefore $\text{AlgLocal}(\mathcal{P})$ does not yield a correct result. ◀

We remark that point sets with the same property exist on any number $n \geq 9$ of points: it suffices to (repeatedly) duplicate the edge qt and perturb its endpoint t .

6 Conclusions

We leave several open questions:

1. What is the correct approximation factor of the algorithm `AlgSimple(\mathcal{P})` presented in Section 2? While each single lemma in Section 2 is tight for some case, it is hard to believe that the whole analysis, leading to the approximation factor $f \doteq 0.5467$, is tight. We conjecture that the algorithm has a better approximation guarantee.
2. What is the approximation factor `BoundDiam(3)` achieved by the polynomial time algorithm that outputs the longest plane tree with diameter 3? By Theorem 5 it is at most $5/6$ (and by [1] it is at least $1/2$).
3. For a fixed $d \geq 4$, is there a polynomial-time algorithm that outputs the longest plane tree with diameter at most d ? By Theorem 6 we know the answer is yes when $d = 3$. And Theorem 7 gives a positive answer for special classes of trees with diameter 4. Note that a hypothetical polynomial-time approximation scheme (PTAS) has to consider trees of unbounded diameter because of Theorem 4. It is compatible with our current knowledge that computing an optimal plane tree of diameter, say, $\mathcal{O}(1/\varepsilon)$ would give a PTAS.
4. Is the general problem of finding the longest plane tree in \mathcal{P} ? A similar question can be asked for several other plane objects, such as paths, cycles, matchings, perfect matchings, or triangulations. The computational complexity in all cases is open.

References

- 1 Noga Alon, Sridhar Rajagopalan, and Subhash Suri. Long non-crossing configurations in the plane. *Fundam. Inform.*, 22(4):385–394, 1995. doi:10.3233/FI-1995-2245.
- 2 Greg Aloupis, Jean Cardinal, Sébastien Collette, Erik D. Demaine, Martin L. Demaine, Muriel Dulieu, Ruy Fabila-Monroy, Vi Hart, Ferran Hurtado, Stefan Langerman, Maria Saumell, Carlos Seara, and Perouz Taslakian. Matching points with things. In Alejandro López-Ortiz, editor, *LATIN 2010: Theoretical Informatics*, volume 6034, pages 456–467, 2010. doi:10.1007/978-3-642-12200-2_40.
- 3 Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.
- 4 Sanjeev Arora and Kevin L. Chang. Approximation schemes for degree-restricted MST and red-blue separation problems. *Algorithmica*, 40(3):189–210, 2004. doi:10.1007/s00453-004-1103-4.
- 5 Alexander I. Barvinok, Sándor P. Fekete, David S. Johnson, Arie Tamir, Gerhard J. Woeginger, and Russell Woodroffe. The geometric maximum traveling salesman problem. *J. ACM*, 50(5):641–664, 2003. doi:10.1145/876638.876640.
- 6 Ahmad Biniiaz. Euclidean bottleneck bounded-degree spanning tree ratios. In *Proc. 31st Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 826–836, 2020. doi:10.1137/1.9781611975994.50.
- 7 Ahmad Biniiaz. Improved approximation ratios for two Euclidean maximum spanning tree problems, 2020. arXiv:2010.03870.
- 8 Ahmad Biniiaz, Prosenjit Bose, Kimberly Crosbie, Jean-Lou De Carufel, David Eppstein, Anil Maheshwari, and Michiel Smid. Maximum plane trees in multipartite geometric graphs. *Algorithmica*, 81(4):1512–1534, 2019. doi:10.1007/s00453-018-0482-x.
- 9 Johannes Blömer. Computing sums of radicals in polynomial time. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci. (FOCS)*, pages 670–677, 1991. doi:10.1109/SFCS.1991.185434.
- 10 Sergio Cabello, Aruni Choudhary, Michael Hoffmann, Katharina Klost, Meghana M Reddy, Wolfgang Mulzer, Felix Schröder, and Josef Tkadlec. A better approximation for longest noncrossing spanning trees. In *36th European Workshop on Computational Geometry (EuroCG)*, 2020.


- 11 Sergio Cabello, Michael Hoffmann, Katharina Klost, Wolfgang Mulzer, and Josef Tkadlec. Long plane trees. *arXiv preprint*, 2021. arXiv:2101.00445.
- 12 Timothy M. Chan. Euclidean bounded-degree spanning tree ratios. *Discrete Comput. Geom.*, 32(2):177–194, 2004. URL: <http://www.springerlink.com/index/10.1007/s00454-004-1117-3>.
- 13 Francis Y. L. Chin, Jianbo Qian, and Cao An Wang. Progress on maximum weight triangulation. In *Proc. 10th Annu. Int. Conf. Computing and Combinatorics (COCOON)*, pages 53–61, 2004. doi:10.1007/978-3-540-27798-9_8.
- 14 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009. URL: <http://mitpress.mit.edu/books/introduction-algorithms>.
- 15 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational geometry. Algorithms and applications*. Springer-Verlag, Berlin, third edition, 2008. doi:10.1007/978-3-540-77974-2.
- 16 Adrian Dumitrescu and Csaba D. Tóth. Long non-crossing configurations in the plane. *Discrete Comput. Geom.*, 44(4):727–752, 2010. doi:10.1007/s00454-010-9277-9.
- 17 Alon Efrat, Alon Itai, and Matthew J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001. doi:10.1007/s00453-001-0016-8.
- 18 David Eppstein. Spanning trees and spanners. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. North Holland / Elsevier, 2000. doi:10.1016/b978-044482537-7/50010-3.
- 19 Andrea Francke and Michael Hoffmann. The Euclidean degree-4 minimum spanning tree problem is NP-hard. In *Proceedings of the 25th ACM Symposium on Computational Geometry*, pages 179–188. ACM, 2009. doi:10.1145/1542362.1542399.
- 20 P. D. Gilbert. New results in planar triangulations. Technical Report R-850, Univ. Illinois Coordinated Science Lab, 1979.
- 21 Sariel Har-Peled. *Geometric approximation algorithms*, volume 173 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2011. doi:10.1090/surv/173.
- 22 Gheza Tom Klincsek. Minimal triangulations of polygonal domains. *Ann. Discrete Math.*, 9:121–123, 1980.
- 23 Joseph S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k -MST, and related problems. *SIAM J. Comput.*, 28(4):1298–1309, 1999. doi:10.1137/S0097539796309764.
- 24 Joseph S. B. Mitchell. Shortest paths and networks. In Jacob E. Goodman and Joseph O’Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 607–641. Chapman and Hall/CRC, 2nd edition, 2004. doi:10.1201/9781420035315.ch27.
- 25 Joseph S. B. Mitchell and Wolfgang Mulzer. Proximity algorithms. In Jacob E. Goodman, Joseph O’Rourke, and Csaba D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 32, pages 849–874. CRC Press, Boca Raton, 3rd edition, 2017. doi:10.1201/9781315119601.
- 26 Wolfgang Mulzer. Minimum dilation triangulations for the regular n -gon. Master’s thesis, Freie Universität Berlin, Germany, 2004.
- 27 Wolfgang Mulzer and Johannes Obenaus. The tree stabbing number is not monotone. In *Proceedings of the 36th European Workshop on Computational Geometry (EWCG)*, pages 78:1–78:8, 2020.
- 28 Wolfgang Mulzer and Günter Rote. Minimum-weight triangulation is NP-hard. *J. ACM*, 55(2):11:1–11:29, 2008. doi:10.1145/1346330.1346336.
- 29 Giri Narasimhan and Michiel Smid. *Geometric spanner networks*. Cambridge University Press, Cambridge, 2007. doi:10.1017/CB09780511546884.
- 30 Christos H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theor. Comput. Sci.*, 4(3):237–244, 1977. doi:10.1016/0304-3975(77)90012-3.

- 31 Christos H. Papadimitriou and Umesh V. Vazirani. On two geometric problems related to the traveling salesman problem. *J. Algorithms*, 5(2):231–246, 1984. doi:10.1016/0196-6774(84)90029-4.
- 32 Jan Remy and Angelika Steger. A quasi-polynomial time approximation scheme for minimum weight triangulation. *J. ACM*, 56(3):15:1–15:47, 2009. doi:10.1145/1516512.1516517.
- 33 Emo Welzl. On spanning trees with low crossing numbers. In *Data structures and efficient algorithms*, volume 594 of *Lecture Notes in Comput. Sci.*, pages 233–249. Springer, Berlin, 1992. doi:10.1007/3-540-55488-2_30.
- 34 David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. doi:10.1017/CB09780511921735.

The Universal ℓ^p -Metric on Merge Trees

Robert Cardona 

University at Albany, State University of New York (SUNY), NY, USA

Justin Curry¹  

University at Albany, State University of New York (SUNY), NY, USA

Tung Lam 

University at Albany, State University of New York (SUNY), NY, USA

Michael Lesnick  

University at Albany, State University of New York (SUNY), NY, USA

Abstract

Adapting a definition given by Bjerkevik and Lesnick for multiparameter persistence modules, we introduce an ℓ^p -type extension of the interleaving distance on merge trees. We show that our distance is a metric, and that it upper-bounds the p -Wasserstein distance between the associated barcodes. For each $p \in [1, \infty]$, we prove that this distance is stable with respect to cellular sublevel filtrations and that it is the universal (i.e., largest) distance satisfying this stability property. In the $p = \infty$ case, this gives a novel proof of universality for the interleaving distance on merge trees.

2012 ACM Subject Classification Mathematics of computing \rightarrow Algebraic topology; Theory of computation \rightarrow Unsupervised learning and clustering; Theory of computation \rightarrow Computational geometry

Keywords and phrases merge trees, hierarchical clustering, persistent homology, Wasserstein distances, interleavings

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.24

Related Version *Full Version:* <https://arxiv.org/abs/2112.12165>

Funding *Justin Curry:* Supported by NSF CCF-1850052 and NASA 80GRC020C0016.

Acknowledgements While Håvard Bjerkevik was not directly involved in this project, he has had a major influence on it, via his collaboration with ML on presentation distances for multiparameter persistence modules [9]. In particular, Håvard kindly agreed to share an early draft of [9] with our group in July 2020, which inspired many of the ideas in our paper.

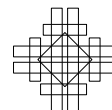
1 Introduction

1.1 Overview

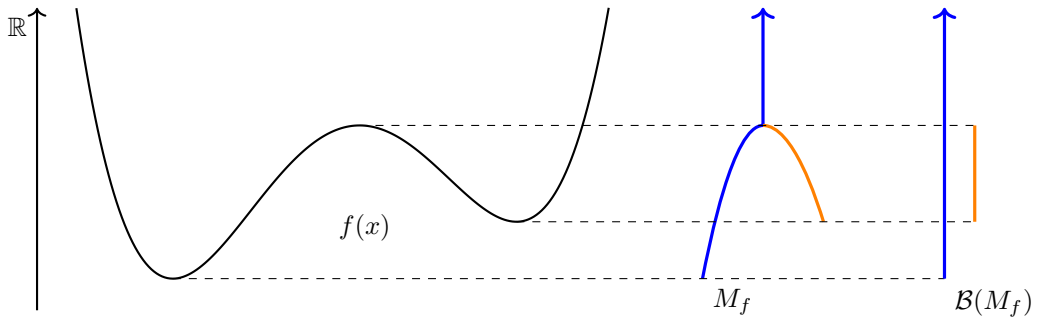
A *merge tree*, also known as a barrier tree [34] or a join tree [18], encodes the connectivity of the sublevel sets of a function $f : X \rightarrow \mathbb{R}$ in terms of a graph M_f equipped with a map $\pi : M_f \rightarrow \mathbb{R}$; see Figure 1. Merge trees are readily computed in practice, and have found applications in topography [39, 36], chemistry [34], visualization [19, 46], cluster analysis [37, 17, 22] and stochastic processes [31, 44, 45]. As a fundamental topological descriptor, merge trees also play a central role in topological data analysis (TDA).

Merge trees are closely related to *persistent homology*, the most widely studied and applied TDA method. Persistent homology provides invariants of data called *barcodes*; a barcode is simply a collection of intervals in \mathbb{R} . Each merge tree M has an associated barcode $\mathcal{B}(M)$,

¹ Corresponding Author



24:2 The Universal ℓ^p -Metric on Merge Trees



■ **Figure 1** Graph of function $f : \mathbb{R} \rightarrow \mathbb{R}$, its associated merge tree M_f and the barcode $\mathcal{B}(M_f)$ obtained from M_f via branch decomposition.

which is obtained via a branch decomposition known as the *elder rule*. This barcode is in fact the same as the *sublevel set* persistence barcode in homological degree 0 considered in TDA [25].

The question of how to metrize the collection of merge trees is a fundamental one: Metrics are needed to study the continuity and stability of the merge tree construction, and to quantify sensitivity to noise. Many metrics on merge trees have been proposed in prior work, as we discuss in detail below.

In particular, metrics called *interleaving distances*, which generalize the well-known *Hausdorff distance* on subsets of a metric space, play a major role in TDA theory. Interleaving distances were first introduced by Chazal et al. [21]; subsequently, the definition has been extended in several different directions [14, 40, 12, 48, 28, 27, 38, 29]. Morozov et al. observed that there is a natural definition of an interleaving distance for merge trees, denoted d_I , and used this to prove the following stability properties of merge trees and their barcodes [41, Theorems 2 and 3]:

► **Theorem 1.1** (Stability properties of merge trees [41]).

(i) For any functions $f, g : X \rightarrow \mathbb{R}$ where X is a topological space, we have that

$$d_I(M_f, N_g) \leq \|f - g\|_\infty, \quad \text{where} \quad \|f - g\|_\infty = \sup_{x \in X} |f(x) - g(x)|.$$

(ii) For any merge trees M and N , we have that

$$d_B(\mathcal{B}(M), \mathcal{B}(N)) \leq d_I(M, N),$$

where d_B denotes the bottleneck distance between barcodes; see Definition 4.9.

While d_B is the most common metric on barcodes in the TDA literature, it has a property that is undesirable in some settings: Informally, d_B is sensitive only to the largest difference between two barcodes and not to smaller differences. To avoid such undesirable behavior, many applications of persistent homology and some theoretical works [23, 52, 49] consider a generalization of d_B called the *p-Wasserstein distance*, denoted $d_{\mathcal{W}}^p$; see Page 14 for the definition. Here $p \in [1, \infty]$ is a parameter and for $p = \infty$ we have that $d_{\mathcal{W}}^\infty = d_B$.² As the

² There are various definitions of the p -Wasserstein distance in the TDA literature, which differ from each other by at most a factor of 2 [23, 13]. In this paper, we use the version introduced by Robinson and Turner [47], and studied in [49, 9].

parameter p decreases, the distances $d_{\mathcal{W}}^p$ become more sensitive to small differences between a pair of barcodes. Because of this, the distances $d_{\mathcal{W}}^p$ with small p , typically $p = 1$ or 2 , are often preferred in practical applications; see [9, Section 1] for a list of applications.

Both the bottleneck distance d_B and the merge tree interleaving distance d_I turn out to be instances of a general categorical definition of interleaving distances introduced by Bubenik and Scott [14]; this is shown for d_B in [6] and for d_I in [26, Proposition 3.11]. As such, the two distances are closely related. It is perhaps unsurprising, then, that the undesirable properties of d_B mentioned above carry over to d_I : That is, d_I is only sensitive to the largest difference between a pair of merge trees, and is insensitive to the smaller differences between them.

With this in mind, it is natural to ask whether we can define an ℓ^p -type distance on merge trees analogous to the distance $d_{\mathcal{W}}^p$ on barcodes, with similar theoretical properties as those given by Theorem 1.1. In this paper, we introduce such a distance, the p -presentation distance, for each $p \in [1, \infty]$. This distance is an analogue of the ℓ^p -distance on multiparameter persistence modules recently introduced in [9], and several of our main results are merge tree analogues of results from [9].

To state an analogue of Theorem 1.1 for presentation distances, we will need some definitions: Let X be a regular cell complex. Following [49], we say $f : X \rightarrow \mathbb{R}$ is *cellular* if it is constant on each cell of X ; and we say f is *monotone* if, in addition, its value on any cell σ is greater than or equal to the values on $\partial\sigma$. Ordering the cells of X arbitrarily, we may identify f with an element of $\mathbb{R}^{|\text{Cells}(X)|}$, so that the ℓ^p -norm $\|f\|_p$ is well defined.

► **Theorem 1.2** (ℓ^p -stability properties of merge trees). *For all $p \in [1, \infty]$,*

(i) *any pair of monotone cellular functions $f, g : X \rightarrow \mathbb{R}$ satisfies*

$$d_I^p(M_f, N_g) \leq \|f - g\|_p,$$

(ii) *any pair of merge trees M and N satisfies*

$$d_{\mathcal{W}}^p(\mathcal{B}(M), \mathcal{B}(N)) \leq d_I^p(M, N).$$

Theorem 1.2 refines the degree-0 case of a fundamental ℓ^p -stability result for persistent homology, due to Skraba and Turner [49]. We also establish the following universality result for d_I^p , which parallels a result on 1- and 2-parameter persistence modules proved in [9].

► **Theorem 1.3.** *For any $p \in [0, \infty]$, if d is any metric on merge trees satisfying the stability property of Theorem 1.2 (i), then $d \leq d_I^p$.*

Several ℓ^∞ -type distances in the TDA literature have been shown to satisfy similar universal properties [40, 10, 5, 3]. In particular, [5] gives a universality result for a metric on Reeb graphs, which are closely related to merge trees.

In addition, we show the following:

► **Theorem 1.4.** $d_I^\infty = d_I$, *i.e., the ∞ -presentation distance and interleaving distance on merge trees are equal.*

Together, Theorems 1.3 and 1.4 give us a universality result for the interleaving distance on merge trees. A version of this result also appears in [8], and was previously announced in a 2019 workshop talk [2]. Whereas our paper only considers merge trees with finitely many nodes, [8] establishes universality of the interleaving distance for locally finite merge trees.

In view of the good theoretical properties of the distances d_I^p , the question of whether these distances can be efficiently computed is interesting. Indeed, if they could be computed, then they could likely be used in practical applications in much the same ways that the

Wasserstein distance on barcodes is commonly used. It is known that computing $d_I = d_I^\infty$ on merge trees is NP hard [1], but is fixed-parameter tractable [33]; we would like to know whether these results extend to d_I^p for all $p < \infty$, but we leave this to future work.

1.2 Other metrics on merge trees

While our p -presentation distance on merge trees is novel, many metrics on merge trees have been considered. Recall that Morozov’s interleaving distance d_I [41], discussed above, is one example. We mention several others: Various forms of edit distances on Merge trees have been proposed [50, 46, 51]. Since merge trees can be viewed as metric spaces in their own right, the *Gromov-Hausdorff distance* on metric spaces can be used to compare merge trees [1]. A Fréchet-like distance between rooted trees was introduced in [32], along with an algorithm to compute this distance. This distance was then applied to merge trees.

The p -cophenetic distance [15] is a metric on labeled merge trees which is similar to d_I^p ; see Definition 5.1. In the $p = \infty$ case, an extension to unlabeled merge trees [42, 35] was shown to be equal to d_I . Consequently, by Theorem 1.4, the ∞ -cophenetic distance and the ∞ -presentation distance are the same. However, for $p < \infty$, d_I^p is a lower bound for the p -cophenetic distance. Example 5.2 illustrates how they differ and demonstrates that the p -cophenetic distance lacks the stability of Theorem 1.2 (i).

In addition, several metrics on *Reeb graphs* have been studied; since the geometric realization (Definition 2.9) of every merge tree is a Reeb graph, any metric on Reeb graphs specializes to a metric on merge trees. A definition of interleavings different from that in [41] was used to define a metric on Reeb graphs in [28]. A family of truncated interleaving distances generalizing this was introduced in [20]. The *functional distortion distance* on Reeb graphs [4] was shown to satisfy stability properties analogous to Theorem 1.1 and to be strongly equivalent to the interleaving distance [7]. Edit distances on Reeb graphs were defined in [30, 5], and [5] showed that its Reeb graph edit distance is universal. Recent work [11] surveys these metrics on Reeb graphs and their relationships. Finally, the *contortion distance* [8] was shown to be strongly equivalent to each of the distances considered in [28], [4], and [5], and to be universal on contour trees.

2 Merge trees

In this section, we define merge trees. We work primarily with a categorical definition, which is convenient for defining the interleaving and presentation distances. Recall that we may regard any partially ordered set (P, \preceq) as a category with one object for each element $p \in P$ and a morphism from p to q whenever $p \preceq q$. We will be particularly interested in the posets \mathbb{R} and $[n] = \{0, 1, \dots, n\}$.

► **Definition 2.1.** *Given a topological space X and function $f: X \rightarrow \mathbb{R}$, the sublevel set filtration of f is the functor $S^\uparrow f: \mathbb{R} \rightarrow \mathbf{Top}$ given by $S^\uparrow f(t) = f^{-1}(-\infty, t]$, with $S^\uparrow f(s \leq t)$ the inclusion $S^\uparrow f(s) \hookrightarrow S^\uparrow f(t)$.*

► **Definition 2.2** (cf. [43, 25]). *A persistent set is a functor $M: \mathbb{R} \rightarrow \mathbf{Set}$.*

► **Example 2.3.** Letting $\pi_0: \mathbf{Top} \rightarrow \mathbf{Set}$ denote the connected components functor, the composition $\pi_0 \circ S^\uparrow f$ is a persistent set.

► **Definition 2.4** (cf. [43]). We say a persistent set M is **constructible** if there exists a set

$$\tau := \{s_0 < s_1 < \dots < s_n\} \subset \mathbb{R} \quad \text{such that}$$

1. If $M \neq \emptyset$, then $\tau \neq \emptyset$ and $M(t) = \emptyset$ for all $t < s_0$,
2. $M(s \leq t)$ is an isomorphism whenever $s, t \in [s_i, s_{i+1})$, and also for $s, t \in [s_n, \infty)$.

If M is constructible, then we call the minimal such τ the set of **critical times** of M , and denote it τ_M . For $s_i \in \tau$, we call $M(s_i)$ a **critical set**.

Note that to specify a constructible merge tree M (up to isomorphism), it suffices to specify τ_M , the critical sets $M(s_i)$, and the functions $m_i := M(s_i \leq s_{i+1})$.

► **Remark 2.5.** Equivalently, one can define constructibility using categorical language: A persistent set M is constructible if it is isomorphic to the left Kan extension of some functor $M' : [n] \rightarrow \mathbf{Set}$ along an order preserving map $j : [n] \hookrightarrow \mathbb{R}$.

► **Definition 2.6.** A **merge forest** is a constructible persistent set M where each $M(t)$ is finite. A **merge tree** is a merge forest where $M(t) = \{*\}$ for t sufficiently large.

We denote the category of merge forests by **Forest** and the category of merge trees by **Merge**. The categorical perspective on merge trees was previously considered in [12, 43, 25] and offers the advantage of a streamlined definition of the interleaving distance; Definition 4.6.

► **Remark 2.7.** Applying the usual disjoint union of sets at each index, we obtain a well defined notion of disjoint union of persistent sets. The disjoint union of finitely many merge forests is itself a merge forest.

The next example is fundamental, as it provides a notion of generators for merge trees.

► **Example 2.8.** A **(closed) strand born at s** , written $F_s : \mathbb{R} \rightarrow \mathbf{Set}$, is the persistent set

$$F_s(t) := \begin{cases} \emptyset & \text{if } t < s, \\ \{*\} & \text{if } t \geq s. \end{cases}$$

Closed strands are clearly constructible. The analogous **open strands**, where $F_s(t) = \{*\}$ if and only if $t > s$, are not constructible. Our strands will always be closed strands. For $m \in \mathbb{Z}_{>0}$, we let F_s^m denote the disjoint union of m copies of F_s .

Following [26], we define the geometric realization of a constructible persistent set; this relates our categorical definition of a merge tree to the topological and graph-theoretic definitions that one typically sees in the literature. Our definition is equivalent to that of [26, Definition A.3] in the constructible setting, though slightly different in the details.

► **Definition 2.9** (Geometric realization). Given a constructible persistent set M , the **geometric realization** of M is a pair $|M| = (X, \gamma)$ where X is a topological space and $\gamma : X \rightarrow \mathbb{R}$ is a continuous function. We take the set underlying X to be $\sqcup_{t \in \mathbb{R}} M_t$, and for $x \in M_t$, we define $\gamma(x) = t$. It remains to specify the topology on X . We regard X as a poset, with $x \preceq y$ iff both $\gamma(x) \leq \gamma(y)$ and $M(\gamma(x) \leq \gamma(y))(x) = y$. For $x \in X$, let

$$C_x = \{y \in X \mid y \text{ and } x \text{ are comparable}\}.$$

We declare a set $U \subset X$ to be open if and only if for each $x \in U$, there exists $V \subset \mathbb{R}$ open such that $(\gamma^{-1}(V) \cap C_x) \subset U$. It is easily verified that this indeed defines a topology on X , and that γ is continuous with respect to this topology.

24:6 The Universal ℓ^p -Metric on Merge Trees

Abusing terminology slightly, we say that a function $f : Y \rightarrow \mathbb{R}$ is **constructible** if $\pi_0 \circ S^\uparrow f$ is constructible. One can check that if f is constructible and continuous, then $|\pi_0 \circ S^\uparrow f|$ is equivalent to the topological construction of a merge tree considered in [41].

A point u is an **ancestor** of a point v if $v \preceq u$. The **least common ancestor** $\text{LCA}(v, w)$ of nodes v and w is the common ancestor of v and w with minimal height. $\text{LCA}(v, w)$ may not exist, but if it exists it is unique.

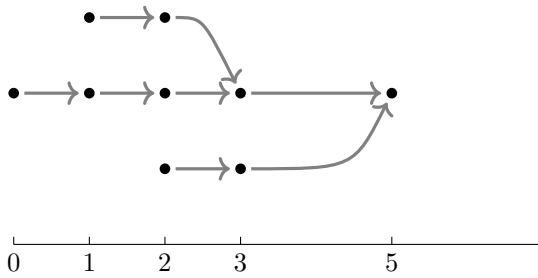
► **Example 2.10.** We specify a merge tree M by taking $\tau_M = \{0, 1, 2, 3, 5\}$, $M_0 := \{0\}$, $M_1 := \{0, 1\}$, $M_2 := \{0, 1, 2\}$, $M_3 = \{0, 2\}$, and $M_5 = \{0\}$,

$$m_2 : \begin{cases} 0 \mapsto 0 \\ 1 \mapsto 0 \\ 2 \mapsto 2 \end{cases} \quad \text{and} \quad m_3 : \begin{cases} 0 \mapsto 0 \\ 2 \mapsto 0 \end{cases}$$

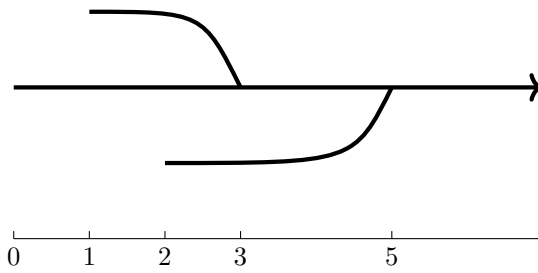
and the remaining m_i to be inclusions. The diagram

$$M_0 \xrightarrow{m_0} M_1 \xrightarrow{m_1} M_2 \xrightarrow{m_2} M_3 \xrightarrow{m_3} M_5$$

can be represented pictorially by:



The geometric realization of M can be drawn as follows:



Barcodes of merge trees

Fix a field k . A **persistence module** is a functor $N : \mathbb{R} \rightarrow \mathbf{Vect}$, where \mathbf{Vect} denotes the category of k -vector spaces. Every persistent set $M : \mathbb{R} \rightarrow \mathbf{Set}$ has an associated persistent homology module $H_0(M) : \mathbb{R} \rightarrow \mathbf{Vect}$ given by the free functor from \mathbf{Set} to \mathbf{Vect} .

Recall that a barcode is a multiset of intervals in \mathbb{R} . According to a well-known structure theorem [24], there is a unique barcode $B(N)$ associated to any pointwise finite dimensional (PFD) persistence module N . Thus, any merge tree M has a well-defined barcode $B(M) := B(H_0(M))$. As suggested in the introduction, $B(M) = B(H_0(S^\uparrow f))$ where $|M| = (X, f)$ is the geometric realization of M . Moreover, if $f : X \rightarrow \mathbb{R}$ is a constructible continuous function, then $B(\pi_0 \circ S^\uparrow f) = B(H_0(S^\uparrow f))$ [25].

3 Presentations of merge trees

We now define presentations of merge trees. Recall that, in the usual algebraic setting, presentations are defined in terms of generators and relations. For merge trees, generators correspond to closed strands, as defined in Example 2.8, which can be thought of as starting branches that emanate from each leaf node. An internal/merge node above leaf nodes i and j witnesses the equality $\text{branch}_i = \text{branch}_j$ by mapping a relating strand into the generating strands for i and j . Each presentation P_M of a merge tree M then gives rise to a matrix that encodes which generators are identified by which relation. We show that any pair of merge trees can be given presentations so that these matrices are identical – we call these “compatible presentations.”

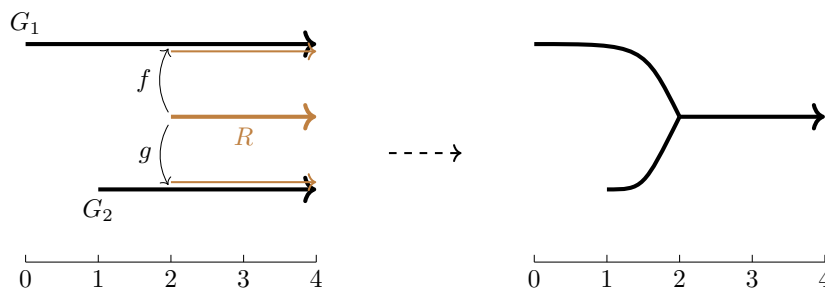
► **Definition 3.1** (Coequalizer). *Given sets A, B , the **coequalizer** of a pair of functions $\alpha, \beta: A \rightrightarrows B$ is the set of equivalence classes $C := B / \sim$, where \sim is the equivalence relation on B generated taking $\alpha(x) \sim \beta(x)$ for all $x \in A$. Let $q: B \rightarrow C$ denote the quotient map. C satisfies the following universal property: Given a function $\gamma: B \rightarrow D$ such that $\gamma \circ \beta = \gamma \circ \alpha$, there is a unique function $\delta: C \rightarrow D$ such that $\gamma = \delta \circ q$. The definition of a coequalizer extends pointwise to persistent sets: Given persistent sets A and B , the **coequalizer** of a pair of natural transformations $\alpha, \beta: A \rightrightarrows B$ is the persistent set C such that for all $t \in \mathbb{R}$, $C(t)$ is the coequalizer of $\alpha(t), \beta(t): A(t) \rightrightarrows B(t)$, with the maps internal to C given by universal properties. The universal property of coequalizers of persistent sets is completely analogous.*

► **Definition 3.2** (Presentation). *A collection of strands $\{G_i\}$ and $\{R_j\}$ – called **generators** and **relations**, respectively – and merge functions $f_j, g_j: R_j \rightarrow G := \sqcup_i G_i$ define a **presentation** of M if M is the coequalizer in **Forest** of the following diagram, written P_M ,*

$$\sqcup_j R_j \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} \sqcup_i G_i \dashrightarrow M.$$

The maps f and g are induced by the merge functions f_j, g_j and the universal mapping property of disjoint unions.

Although this definition is cloaked in category theory, coequalizers and disjoint unions formalize gluing constructions in topology, which are more generally cast in terms of colimits. Intuitively, relating strands indicate where generating strands are glued together; see Figure 2.



■ **Figure 2** Presenting a merge tree with two branches as a coequalizer.

We can encode a presentation by the presentation matrix and its label vector.

24:8 The Universal ℓ^p -Metric on Merge Trees

► **Definition 3.3** (Presentation matrix and label vector). Given a presentation P_M of a merge tree M with k generators and l relations,

$$\bigsqcup_{j=1}^l R_j \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} \bigsqcup_{i=1}^k G_i \dashrightarrow M,$$

we can pick an ordering of generators and relations to obtain a $(k \times l)$ **presentation matrix** where the i -th row corresponds to the i -th generator G_i and the j -th column corresponds to the j -th relation R_j . The (i, j) -entry of the presentation matrix is 1 if either of $f_j, g_j : R_j \rightarrow G$ maps to G_i and 0 otherwise.

The **label vector** of P_M is the $(k + l)$ -vector $L(P_M)$ where the first k entries encode the birth times of each generating strand and the remaining l entries encode the birth times of each of the relating strands. We will separate the row (generator) birth times from the column (relation) birth times by a semi-colon for legibility.

We now give several examples of presentations, presentation matrices, and their label vectors. We will see in particular that presentations are *not* unique. Indeed, one can always modify an existing presentation by introducing an extra generating strand that is then killed by an identical relating strand, as in Examples 3.4 and 3.5. Example 3.6 shows that once two generators have been merged, any further merge event can be encoded by a merge function that maps to either generator.

► **Example 3.4** (Tree with one leaf node). If M is a merge tree with one leaf node born at time s , then there is a presentation P_M given by

$$F_s \begin{array}{c} \xrightarrow{\text{id}} \\ \xrightarrow{\text{id}} \end{array} F_s \dashrightarrow M.$$

The corresponding presentation matrix and label vector are, respectively,

$$s \quad (1) \quad \text{and} \quad L(P_M) = [s; s].$$

One can also obtain a presentation P'_M of M with one generator F_s and no relations. In this case, the presentation matrix of P'_M is an (empty) 1×0 matrix, but whose label vector is $L(P'_M) = [s]$.

► **Example 3.5** (Adding a trivial generator and relation). Let M be a merge tree with two leaves born at times 0 and 1 that merge at time 2. The presentation P_M in Figure 2 uses two generators and one relation

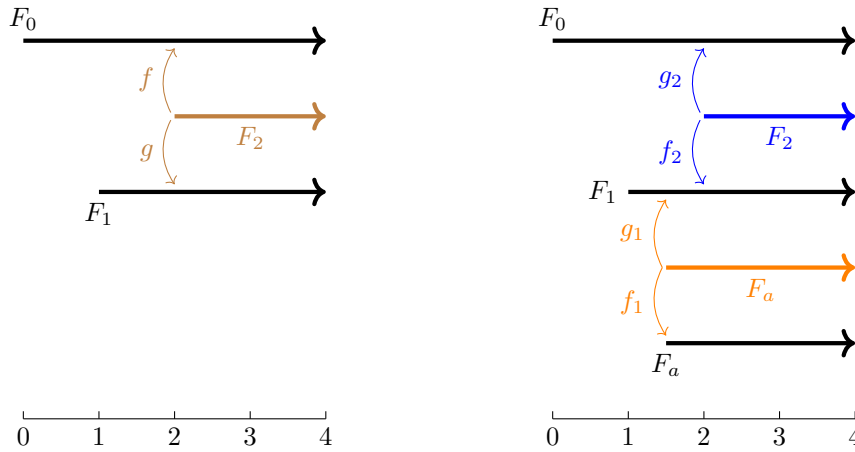
$$F_2 \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} F_0 \sqcup F_1 \dashrightarrow M,$$

where $f : F_2 \rightarrow F_0$, and $g : F_2 \rightarrow F_1$ are Merge functions. One can modify P_M to obtain a presentation P'_M by introducing an extra generator and relation F_a , for any $a \in [1, 2)$,

$$F_a \sqcup F_2 \begin{array}{c} \xrightarrow{f_1 \sqcup f_2} \\ \xrightarrow{g_1 \sqcup g_2} \end{array} F_0 \sqcup F_1 \sqcup F_a \dashrightarrow M,$$

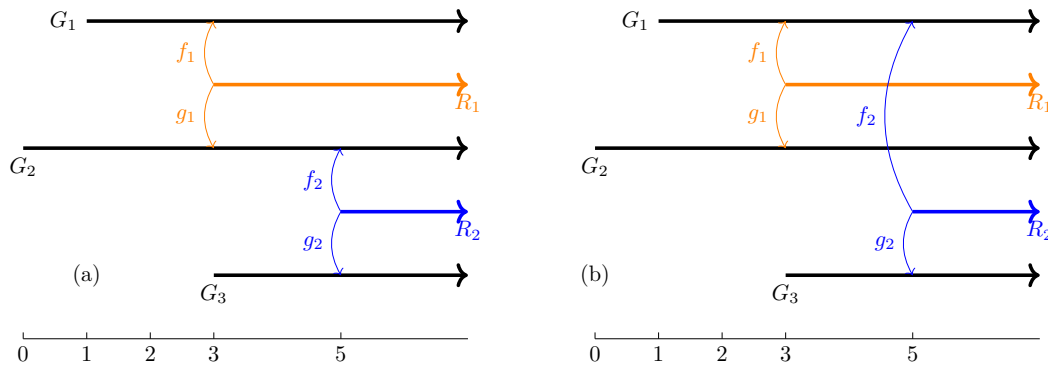
with $f_1 : F_a \rightarrow F_a$, $g_1 : F_a \rightarrow F_1$ and $f_2 : F_2 \rightarrow F_1$, $g_2 : F_2 \rightarrow F_0$; see Figure 3. The corresponding presentation matrices and label vectors for P_M and P'_M are then, respectively,

$$\begin{matrix} & 2 \\ 0 & \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{matrix} L(P_M) = [0, 1; 2] \quad \text{and} \quad \begin{matrix} & 2 & a \\ 0 & \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \end{matrix} L(P'_M) = [0, 1, a; 2, a].$$



■ **Figure 3** Presentations P_M (left) and P'_M (right) of the merge tree from Figure 2.

► **Example 3.6** (Different merge functions). Consider the merge tree from Example 2.10 with three leaf nodes and two least common ancestors. Since one least common ancestor occurs after the other, there is a choice as to which generator you choose to attach to. Figure 4 shows these two possible choices. The corresponding presentation matrices for these are



■ **Figure 4** Two different presentations for the merge tree introduced in Example 2.10. R_2 can be matched with either G_1 or G_2 since after R_1 they have already merged into one component.

$$\begin{matrix} & 3 & 5 \\ 1 & \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \end{matrix} \quad \text{and} \quad \begin{matrix} & 3 & 5 \\ 1 & \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{matrix},$$

but both of these presentations have the same label vector $L = [1, 0, 3; 3, 5]$.

24:10 The Universal ℓ^p -Metric on Merge Trees

We now present two key lemmas.

► **Lemma 3.7.** *Every merge tree with n leaf nodes has a presentation with n generators and $n - 1$ relations.*

Proof. We proceed by induction on the number of leaf nodes: Example 3.5 shows that the claim holds for $n = 2$. Suppose that the claim is true for some $k > 2$ and M is a merge tree of $k + 1$ leaf nodes. Let t be the highest merge time of M , there exists m groups of $k_1, k_2, \dots, k_m > 0$ leaf nodes whose merge times are not larger than t . Notice that each i -th group of k_i leaf nodes can be realized as a merge tree whose presentation consists of k_i generators and $k_i - 1$ relations. Let G_i be a representative strand of the i -th group, ($1 \leq i \leq m$), by pairwise relating G_1 and G_i at the time t ($2 \leq i \leq m$), one obtains a presentation that consists of $k + 1$ generators. The number of relations is given by $(k_1 - 1) + \dots + (k_m - 1) + (m - 1)$, which is k . ◀

We say that two presentations P_M and P_N are **compatible** if their underlying matrices \bar{P}_M and \bar{P}_N are the same.

► **Lemma 3.8.** *Any pair of merge trees M and N have compatible presentations.*

Proof. Given presentations of M and N , we may add extra generators and relations to one of them, to obtain presentations P_M and P_N for M and N , respectively, with the same number of generators. Write the matrices underlying P_M and P_N as \bar{P}_M and \bar{P}_N , respectively, and denote their numbers of relations by m and n . Since M and N are merge trees, there exists $t \in \mathbb{R}$ such that $M_{t'}$ and $N_{t'}$ are singleton sets for all $t' \geq t$. We construct compatible presentations for \tilde{P}_M and \tilde{P}_N with underlying matrix $(\bar{P}_M \quad \bar{P}_N)$: For \tilde{P}_M , we take the row labels and the first m column labels to be the same as for P_M , with each of the last n column labels equal to t ; and for \tilde{P}_N , we take the row labels and the last n column labels be the same as for P_N , with each of first m column labels equal to t . Since at time t all the strands of M have been related to each other, \tilde{P}_M is indeed a presentation for M . Similarly, \tilde{P}_N is a presentation for N . ◀

► **Remark 3.9.** More generally, two merge forests have compatible presentations if and only if they have the same number of connected components.

4 Presentation metric on merge trees

We next introduce the p -presentation metrics merge trees, adapting the definitions of [9]. We first define a semi-metric on merge-trees (Definition 4.1) which measures the difference between merge trees in terms of the ℓ^p -distance between the birth times of the generators and relations in a compatible presentation of M and N . Unfortunately, as Example 4.2 shows, Definition 4.1 fails to satisfy the triangle inequality, so we pass to sequences of merge trees in Definition 4.3 to get a genuine (pseudo)metric.

► **Definition 4.1** (p -presentation semi-distance). *If P_M and P_N are compatible presentations for merge trees M and N , then for any $p \in [1, \infty]$ we define the **p -label distance** to be $\|L(P_M) - L(P_N)\|_p$. The **p -presentation semi-distance** between merge trees M and N is*

$$\hat{d}_1^p(M, N) = \inf\{\|L(P_M) - L(P_N)\|_p \mid P_M \text{ and } P_N \text{ are compatible}\}.$$

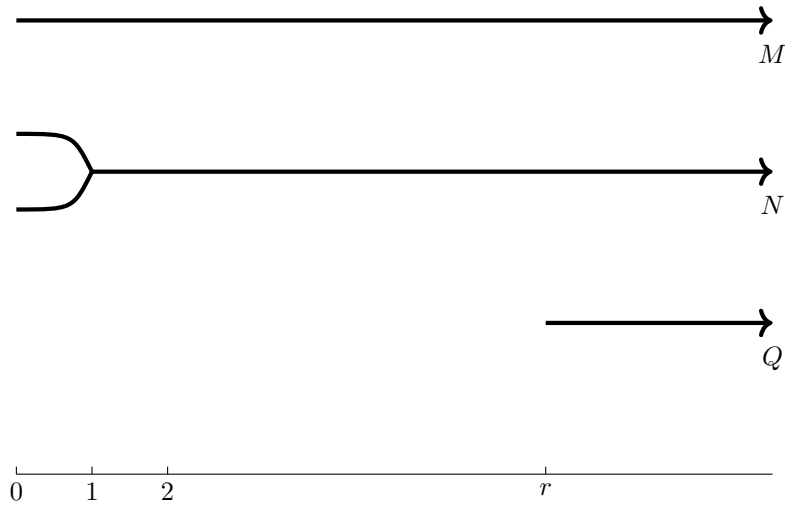


Figure 5 Counterexample to the triangle inequality for \hat{d}_I^p .

Example 4.2. The following is a close analogue of [9, Example 3.1]. In Figure 5 we have three merge trees M, N and Q . We claim that for r large enough

$$\hat{d}_I^p(M, N) \leq 1, \quad \hat{d}_I^p(M, Q) = r, \quad \text{and} \quad \hat{d}_I^p(N, Q) = \sqrt[p]{(r-1)^p + 2r^p}.$$

From this it follows that

$$\hat{d}_I^p(N, Q) \geq \sqrt[p]{(r-1)^p + 2r^p} > 1 + r \geq \hat{d}_I^p(N, M) + \hat{d}_I^p(M, Q),$$

and hence the triangle inequality does not hold for \hat{d}_I^p .

Consider the following compatible presentations

$$P_M := \begin{matrix} & \epsilon & \\ 0 & \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \\ \epsilon & & \end{matrix}, \quad P_N := \begin{matrix} & 1 & \\ 0 & \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \\ 0 & & \end{matrix}, \quad P_Q := \begin{matrix} & & r + \epsilon \\ r & \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \\ r + \epsilon & & \end{matrix}.$$

It is easy to see that $\hat{d}_I^p(M, Q) = r$ by choosing compatible presentations with a single generator and no relations. To see that $\hat{d}_I^p(M, N) \leq 1$, observe that

$$\hat{d}_I^p(M, N) \leq \|L(P_M) - L(P_N)\|_p = \sqrt[p]{1 - \epsilon^p + \epsilon^p}.$$

Setting $\epsilon = 0$ for the presentation of Q shows that $\hat{d}_I^p(N, Q) \leq \sqrt[p]{(r-1)^p + 2r^p}$. Any pair of compatible presentations P'_N and P'_Q for N and Q must contain a subpresentation of the form P_Q , since P_N is minimal. However, for all $\epsilon \geq 0$, the label vector difference $\|L(P'_M) - L(P'_N)\|_p$ will be greater than $\sqrt[p]{r^p + (r + \epsilon)^p + (r + \epsilon - 1)^p}$, which is minimized at $\epsilon = 0$.

Although Example 4.2 shows that \hat{d}_I^p is not a metric, we can repair this as follows:

Definition 4.3 (p -presentation distance). For merge trees M and N and $p \in [1, \infty]$ we define the p -presentation distance as

$$d_I^p(M, N) := \inf \sum_{i=0}^{n-1} \hat{d}_I^p(Q_i, Q_{i+1}),$$

where we infimize over all finite sequences of merge trees $M = Q_0, \dots, Q_n = N$.

24:12 The Universal ℓ^p -Metric on Merge Trees

The following result is a close analogue of [9, Proposition 3.6]. It has essentially the same proof and will play a similarly important role in our arguments.

► **Lemma 4.4** (Largest bounded metric).

- (i) d_I^p is a metric on isomorphism classes of merge trees.
- (ii) d_I^p is the largest such metric bounded above by \hat{d}_I^p .

Proof. By Lemma 3.8, the presentation distance between any two merge trees is finite. It then follows from [9, Rmk. 3.4], which is reproduced as Lemma A.1 in the full version of this paper [16], that d_I^p is the largest pseudometric bounded above by \hat{d}_I^p . Finally, Lemma A.2 in the full version shows that if $d_I^p(M, N) = 0$, then M is isomorphic to N , which finishes the proof. ◀

To formalize how varying p increases the sensitivity of this distance, we recall the fundamental property of ℓ^p -norms: for any vector x , $\|x\|_p \geq \|x\|_q$ whenever $p \leq q$.

► **Proposition 4.5.** For any pair of merge trees M and N and for all $1 \leq p \leq q \leq \infty$, we have $d_I^p(M, N) \geq d_I^q(M, N)$.

4.1 Equality of the ∞ -presentation distance and interleaving distance

We now prove Theorem 1.4, which says that the ∞ -presentation distance is equal to the interleaving distance. First, we define the interleaving distance.

► **Definition 4.6.** For $\epsilon > 0$, there is a shift functor $S_\epsilon : \mathbb{R} \rightarrow \mathbb{R}$ given by $t \mapsto t + \epsilon$. An ϵ -**interleaving** between persistent sets M and N is given by a pair of natural transformations $\varphi : M \rightarrow NS_\epsilon$ and $\psi : N \rightarrow MS_\epsilon$ so that the following diagrams commute for all $s \in \mathbb{R}$,

$$\begin{array}{ccccc} M(s) & \longrightarrow & M(s + \epsilon) & \longrightarrow & M(s + 2\epsilon) \\ & \searrow & \nearrow & \searrow & \nearrow \\ N(s) & \longrightarrow & N(s + \epsilon) & \longrightarrow & N(s + 2\epsilon) \end{array}$$

The **interleaving distance** between two persistent sets M and N , and hence two merge trees, is defined as

$$d_I(M, N) := \inf\{\epsilon \mid M \text{ and } N \text{ are } \epsilon\text{-interleaved}\}.$$

Proof of Theorem 1.4. Let M and N be two merge trees. By Lemma 4.4(ii) it suffices to prove that $\hat{d}_I^\infty = d_I$ because d_I satisfies the triangle inequality and \hat{d}_I^∞ is the largest pseudometric bounded above by \hat{d}_I^∞ .

Suppose there exists an ϵ -interleaving between M and N : $\varphi : M \rightarrow NS_\epsilon$ and $\psi : N \rightarrow MS_\epsilon$. Let \bar{P}_M (resp. \bar{P}_N) be the underlying matrix for a presentation of M (resp. N), whose generators and relations are G_M and R_M (resp. G_N and R_N). Here we slightly abuse notation by using generators and relations as row and column labels. We define a matrix P_Z as follows,

$$P_Z := \begin{array}{c} G_M \\ G_N \end{array} \begin{pmatrix} \bar{P}_M & 0 \\ 0 & \bar{P}_N \end{pmatrix} \begin{array}{c} R_M \\ R_N \\ P_\varphi \\ P_\psi \end{array} \begin{array}{c} G_M S_\epsilon \\ G_N S_\epsilon \\ I \\ I \end{array}$$

where P_ψ is defined by

$$(i, j) \mapsto \begin{cases} 1 & \text{if } \psi(\pi G_i^N)(g_i^N) = g_j^M, \\ 0 & \text{otherwise,} \end{cases}$$

and P_φ is defined by

$$(i, j) \mapsto \begin{cases} 1 & \text{if } \varphi(\pi G_i^M)(g_i^M) = g_j^N, \\ 0 & \text{otherwise.} \end{cases}$$

Here g_i^N represents the element of $N(\pi G_i^N)$ representing the strand G_i^N . Moreover $G_M S_\epsilon$ denotes the collection of strands in G_M that are obtained by ϵ -shifting, likewise for $G_N S_\epsilon$. By construction, P_ψ and P_φ have exactly one nonzero entry per column. Using this matrix P_Z , one can obtain a pair of compatible presentations P_M and P_N for M and N respectively by relabeling P_Z as follows:

$$P_M := \begin{matrix} & R_M & R_N S_\epsilon & G_M S_{2\epsilon} & G_N S_\epsilon \\ G_M & \left(\begin{array}{cccc} \bar{P}_M & 0 & I & P_\psi \\ 0 & \bar{P}_N & P_\varphi & I \end{array} \right) & & & \\ G_N S_\epsilon & & & & \end{matrix},$$

and

$$P_N := \begin{matrix} & R_M S_\epsilon & R_N & G_M S_\epsilon & G_N S_{2\epsilon} \\ G_M S_\epsilon & \left(\begin{array}{cccc} \bar{P}_M & 0 & I & P_\psi \\ 0 & \bar{P}_N & P_\varphi & I \end{array} \right) & & & \\ G_N & & & & \end{matrix}.$$

Since each of the generator and relation birth times for P_M and P_N differ exactly by ϵ , we can conclude that $\|L(P_M) - L(P_N)\|_\infty = \epsilon$. Hence we have shown that $d_I(M, N) \geq \hat{d}_I^\infty(M, N)$.

Now suppose that there exists a pair of compatible presentations P_M and P_N for M and N respectively, such that $\|L(P_M) - L(P_N)\|_\infty = \epsilon$. This hypothesis guarantees generators of P_M are born within ϵ of generators of P_N , and likewise for their relations. This allows us to construct functorial mappings:

$$\begin{array}{ccccc} \sqcup R_i^M & \rightrightarrows & \sqcup G_i^M & \longrightarrow & M \\ \downarrow \alpha & & \downarrow \beta & & \downarrow \varphi \\ \sqcup R_i^N S_\epsilon & \rightrightarrows & \sqcup G_i^N S_\epsilon & \longrightarrow & N S_\epsilon \end{array}$$

where α and β are defined by taking $G_i^M \mapsto G_i^N S_\epsilon$ and $R_j^M \mapsto R_j^N S_\epsilon$. Here $\varphi : M \rightarrow N S_\epsilon$ is obtained via the universal property of the coequalizer. Commutativity of the left square in the diagram follows from the fact that P_M and P_N have the same underlying matrix. Similarly, we obtain a map $\psi : N \rightarrow M S_\epsilon$. The uniqueness of the universal property guarantees (φ, ψ) to be an interleaving pair. This shows that there exists a ϵ -interleaving between M and N , that is, $d_I(M, N) \leq \hat{d}_I^\infty(M, N)$. ◀

4.2 Comparison with Wasserstein distance on barcodes

Now that metric properties of the p -presentation distance d_I^p have been established, we turn our attention to the relation between this distance and the p -Wasserstein distance d_W^p on barcodes. Specifically, we prove Theorem 1.2 (ii), which says that for $p \in [1, \infty]$ and any merge trees M and N , we have

$$d_W^p(\mathcal{B}(M), \mathcal{B}(N)) \leq d_I^p(M, N).$$

24:14 The Universal ℓ^p -Metric on Merge Trees

► **Lemma 4.7.** For merge trees M and N ,

$$d_I^p(M, N) \geq d_I^p(H_0(M), H_0(N)),$$

where $d_I^p(H_0(M), H_0(N))$ denotes the ℓ^p -distance on persistent modules, as defined in [9].

Proof. Let M and N be merge trees. We first show that $\hat{d}_I^p(M, N) \geq \hat{d}_I^p(H_0(M), H_0(N))$, where $\hat{d}_I^p(H_0(M), H_0(N))$ denotes the p -presentation semi-distance on persistent modules, as defined in [9]. If P_M, P_N are compatible presentations for M and N ,

$$\bigsqcup_i R_i^M \xrightarrow[g^M]{f^M} \bigsqcup_j G_j^M \dashrightarrow M, \quad \text{and} \quad \bigsqcup_i R_i^N \xrightarrow[g^N]{f^N} \bigsqcup_j G_j^N \dashrightarrow N,$$

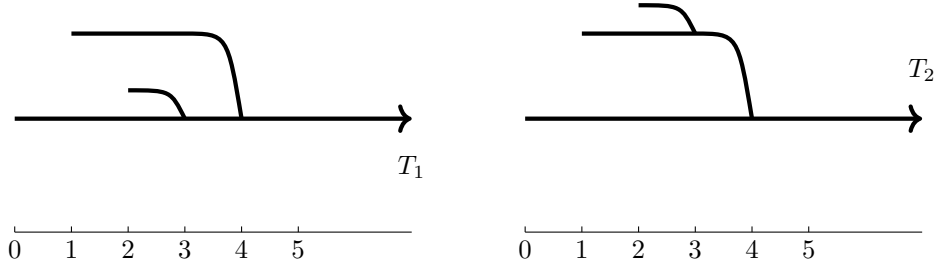
applying H_0 will yield compatible presentations for $H_0(M)$ and $H_0(N)$,

$$\bigoplus_i H_0(R_i^M) \xrightarrow{f^{M*} - g^{M*}} \bigoplus_j H_0(G_j^M) \dashrightarrow H_0(M),$$

$$\bigoplus_i H_0(R_i^N) \xrightarrow{f^{N*} - g^{N*}} \bigoplus_j H_0(G_j^N) \dashrightarrow H_0(N),$$

whose p -distance is $\|L(P_M) - L(P_N)\|_p$. This implies that $\hat{d}_I^p(M, N) \geq \hat{d}_I^p(H_0(M), H_0(N))$. This is sufficient to prove the statement because we know that $\hat{d}_I^p(H_0(M), H_0(N)) \geq d_I^p(H_0(M), H_0(N))$, by [9, Prop. 3.3]. Since $d_I^p(H_0(\bullet), H_0(\bullet))$ is a pseudometric on merge trees, Lemma 4.4(ii) implies that $d_I^p(M, N) \geq d_I^p(H_0(M), H_0(N))$. ◀

► **Remark 4.8.** The inequality from Lemma 4.7 can be strict because non-isomorphic merge trees can have isomorphic persistent homology modules. This was demonstrated in [25], where the following example was considered.



Continuing our comparison of the p -presentation distance on merge trees with metrics in persistent homology, we recall the definition of the Wasserstein distance on barcodes.

► **Definition 4.9.** A **barcode** \mathcal{B} is a finite collection of intervals $\{I\}$ in \mathbb{R} . A **matching** between barcodes \mathcal{B} and \mathcal{C} consists of a choice of subsets $\mathcal{B}' \subset \mathcal{B}$, and $\mathcal{C}' \subset \mathcal{C}$ and a bijection $\sigma : \mathcal{B}' \rightarrow \mathcal{C}'$. For any $p \in [1, \infty]$ we define the **p -cost** of σ as

$$\text{cost}(\sigma, p) = \left\{ \begin{array}{l} \left(\sum_{\substack{I \in \mathcal{B}, \\ \sigma(I) = J}} \|I - J\|_p^p + \sum_{I \in \Delta} \|I - \text{mid}(I)\|_p^p \right)^{1/p}, \text{ when } 1 \leq p < \infty \\ \max \left\{ \max_{\substack{I \in \mathcal{B}, \\ \sigma(I) = J}} \|I - J\|_\infty, \max_{I \in \Delta} \|I - \text{mid}(I)\|_\infty \right\}, \text{ when } p = \infty, \end{array} \right\},$$

where $\|I - J\|_p$ is the ℓ^p -norm between intervals $I = [a, b]$ and $J = [c, d]$ viewed as vectors (a, b) and (c, d) in \mathbb{R}^2 , $\text{mid}(I) := [\frac{a+b}{2}, \frac{a+b}{2}]$ is the empty interval at the midpoint of I , and Δ denotes all the intervals unmatched by σ in $\mathcal{B} \sqcup \mathcal{C}$. The **Wasserstein p -distance** between barcodes \mathcal{B} and \mathcal{C} is then defined as the infimum of p -costs over all possible matchings, i.e.,

$$d_{\mathcal{W}}^p(\mathcal{B}, \mathcal{C}) = \inf_{\sigma} \text{cost}(\sigma, p).$$

The distance $d_{\mathcal{W}}^{\infty}$ is called the **bottleneck distance**.

Lemma 4.7 and [9, Theorem 1.1] together establish Theorem 1.2 (ii).

5 Stability and universality

In this section we consider two of the most important properties of the p -presentation metric on merge trees: stability and universality. To motivate stability, we consider another matrix-based distance on merge trees known as the p -cophenetic distance, which we show is *not* stable for $p \in [1, \infty)$.

Comparison with cophenetic distances

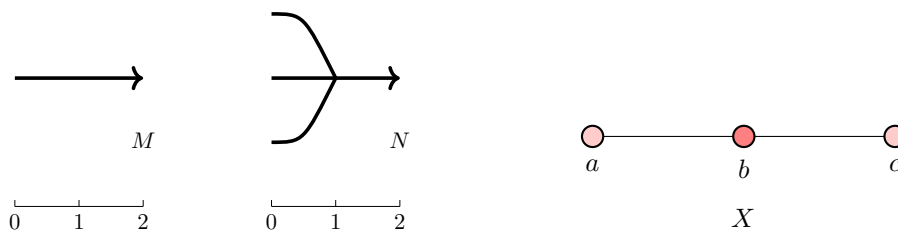
► **Definition 5.1.** Given a merge tree M together with a surjective ordered leaf node labelling $\pi = \{G_1, \dots, G_k\}$, a **cophenetic vector** C_M^{π} is an upper triangular matrix whose (i, j) -entry is the earliest merge time (height of the least common ancestor) of nodes G_i and G_j . For a pair of merge trees M and N together with a labelling π , the **labeled p -cophenetic distance** is defined as $\|C_M^{\pi} - C_N^{\pi}\|_p$, where we view these matrices as length $k(k+1)/2$ -vectors. One can then define the **p -cophenetic distance** between two merge trees as

$$d_C^p(M, N) = \inf_{\pi \in \Pi} \|C_M^{\pi} - C_N^{\pi}\|_p,$$

where Π denotes the set of all surjective, ordered leaf labelings of M and N .

We now give a counterexample to stability of the cophenetic p -distance, when $p \neq \infty$, for cellular monotone functions; see Page 3, Section 1.1 for a reminder.

► **Example 5.2.** Let X be the barycentric subdivision of the geometric 1-simplex. Consider the monotone cellular functions $f, g : X \rightarrow \mathbb{R}$ where $f \equiv 0$ and g is 0 on 0-cells and 1 on 1-cells. By inspection, the merge tree $M = \pi_0 S^{\uparrow} f$ has one leaf node and $N = \pi_0 S^{\uparrow} g$ has three leaf nodes, one for each 0-cell, and a single internal node; see Figure 6.



► **Figure 6** Two merge trees are shown at left, associated to two functions on the cell complex X , at right. In Example 5.2 these give a counterexample to stability of the p -cophenetic distance.

By allowing redundant labels for the one leaf node in M , we have cophenetic vectors

$$C_M = \begin{bmatrix} 0 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix} \quad \text{and} \quad C_N = \begin{bmatrix} 0 & 1 & 1 \\ * & 0 & 1 \\ * & * & 0 \end{bmatrix}.$$

24:16 The Universal ℓ^p -Metric on Merge Trees

From [35] we know that $d_C^\infty(M_f, M_g) = d_I(M_f, M_g)$, which is stable [41], i.e.,

$$d_C^\infty(M_f, M_g) = d_I(M_f, M_g) \leq \|f - g\|_\infty.$$

However, for $p \in [1, \infty)$, we have $d_C^p(M_f, M_g) = \sqrt[p]{3}$, which is larger than $\|f - g\|_p = \sqrt[p]{2}$.

5.1 Stability

► **Definition 5.3.** A distance d on merge trees is said to be **p -stable** if whenever f and g are monotone cellular functions on a regular cell complex, the associated merge trees $M = \pi_0 S^\uparrow f$ and $N = \pi_0 S^\uparrow g$ satisfy $d(M, N) \leq \|f - g\|_p$.

We show that p -presentation distance is p -stable.

Proof of Theorem 1.2(i). We start by labeling vertices of X by $\sigma_1, \dots, \sigma_k$ and edges by τ_1, \dots, τ_l . Consider the $(k \times l)$ -matrix P where

$$P(i, j) := \begin{cases} 1 & \text{if } \sigma_i \subseteq \tau_j, \\ 0 & \text{otherwise.} \end{cases}$$

We then define labeled matrices P_M, P_N with underlying matrix P ,

$$P_M := \begin{matrix} & \rho_1^M & \cdots & \rho_l^M \\ \gamma_1^M & & & \\ \vdots & & & \\ \gamma_k^M & & & \end{matrix} \left(\begin{matrix} & & & \\ & & & \\ & & & \\ & & & \end{matrix} \right) \quad \text{and} \quad P_N := \begin{matrix} & \rho_1^N & \cdots & \rho_l^N \\ \gamma_1^N & & & \\ \vdots & & & \\ \gamma_k^N & & & \end{matrix} \left(\begin{matrix} & & & \\ & & & \\ & & & \\ & & & \end{matrix} \right),$$

where $\gamma_i^M = f(\sigma_i)$ and $\rho_j^M = f(\tau_j)$, likewise for P_N . By definition of a regular cell complex each column of P_M and P_N must have exactly two ones. Moreover, P_M and P_N are compatible presentation matrices of their respective merge trees: Monotonicity guarantees that we can obtain presentations for merge trees M, N with the generators and relations as described in P_M and P_N . By construction we have that $\|L(P_M) - L(P_N)\|_p \leq \|f - g\|_p$ and by definition of \hat{d}_I^p it follows that $d_I^p(M, N) \leq \hat{d}_I^p(M, N) \leq \|f - g\|_p$. ◀

► **Example 5.4.** If we present the merge trees M and N from Example 5.2 using three generators and two relations, the corresponding presentation matrices are

$$P_M := \begin{matrix} & 0 & 0 \\ 0 & \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \end{matrix} \quad \text{and} \quad P_N := \begin{matrix} & 1 & 1 \\ 0 & \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \end{matrix}$$

and the label vectors are $L(P_M) := [0, 0, 0; 0, 0]$ and $L(P_N) := [0, 0, 0; 1, 1]$. Thus the p -label distance is $\sqrt[p]{2}$, which equals the ℓ^p -distance between f and g . Consequently, $\hat{d}_I^p(M_f, N_g) \leq \sqrt[p]{2} = \|f - g\|_p$. Hence, $d_I^p(M_f, N_g) \leq \|f - g\|_p$, thus illustrating one advantage of this metric over the cophenetic p -distance.

5.2 Universality

In this section we prove that the p -presentation metric is universal among p -stable metrics. Here “universal” can be interpreted as maximal or final in a certain poset category of pseudometrics. In order to prove Theorem 1.3, we need the following lemma:

► **Lemma 5.5** (Geometric lifting). *If M and N are merge trees with compatible presentations P_M and P_N such that $\|L(P_M) - L(P_N)\|_p = \epsilon$, then there exists a regular cell complex X and monotone cellular functions $f : X \rightarrow \mathbb{R}$ and $g : X \rightarrow \mathbb{R}$ such that*

- (i) $M \cong \pi_0 S^\uparrow f$, $N \cong \pi_0 S^\uparrow g$, and
- (ii) $\|f - g\|_p = \epsilon$.

Proof. Let P_M and P_N be compatible presentations for M and N with presentation matrices

$$P_M := \begin{matrix} & \rho_1^M & \cdots & \rho_i^M \\ \gamma_1^M & & & \\ \vdots & & & \\ \gamma_k^M & & & \end{matrix} \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix} \quad \text{and} \quad P_N := \begin{matrix} & \rho_1^N & \cdots & \rho_i^N \\ \gamma_1^N & & & \\ \vdots & & & \\ \gamma_k^N & & & \end{matrix} \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}.$$

Using the underlying matrix for either presentation we construct a cell complex X as follows: For each row i , add a 0-cell σ_i to X , and for each column j , attach a 1-cell τ_j so that $P_M(i, j) = 1 \Rightarrow \sigma_i \subseteq \tau_j$. We then construct cellular functions $f, g : X \rightarrow \mathbb{R}$ using the birth times for the generators and relations for P_M and P_N , respectively. More precisely, $f(\sigma_i) = \gamma_i^M$ and $f(\tau_j) = \rho_j^M$, likewise for g . Since $\|L(P_M) - L(P_N)\|_p = \epsilon$ we have that $\|f - g\|_p = \epsilon$ as well. ◀

Proof of Theorem 1.3. It suffices to prove that for any p -stable distance d , we have $d \leq \hat{d}_T^p$. Let M, N be merge trees with $\hat{d}_T^p(M, N) = \epsilon$ and let $\epsilon' > \epsilon$ be arbitrary. By Lemma 5.5, we can find $f, g : X \rightarrow \mathbb{R}$ such that $M \cong \pi_0 S^\uparrow f$, $N \cong \pi_0 S^\uparrow g$, and $\|f - g\|_p = \epsilon'$. By assumption $d(M, N) \leq \|f - g\|_p = \epsilon'$ and by letting $\epsilon' \rightarrow \epsilon$, we have that $d(M, N) \leq \hat{d}_T^p(M, N)$. By Lemma 4.4, we know that $d \leq d_T^p$. ◀

References



- 1 Pankaj K Agarwal, Kyle Fox, Abhinandan Nath, Anastasios Sidiropoulos, and Yusu Wang. Computing the Gromov-Hausdorff distance for metric trees. *ACM Transactions on Algorithms (TALG)*, 14(2):1–20, 2018. doi:10.1145/3185466.
- 2 Ulrich Bauer. The space of Reeb graphs. *Workshop on Topology, Computation, and Data Analysis*, Schloss Dagstuhl, 2019.
- 3 Ulrich Bauer, Magnus Bakke Botnan, and Benedikt Fluhr. Universality of the bottleneck distance for extended persistence diagrams. *arXiv preprint*, 2020. arXiv:2007.01834.
- 4 Ulrich Bauer, Xiaoyin Ge, and Yusu Wang. Measuring distance between Reeb graphs. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pages 464–473, 2014. doi:10.1145/2582112.2582169.
- 5 Ulrich Bauer, Claudia Landi, and Facundo Mémoli. The Reeb graph edit distance is universal. *Foundations of Computational Mathematics*, pages 1–24, 2020. doi:10.1007/s10208-020-09488-3.
- 6 Ulrich Bauer and Michael Lesnick. Persistence diagrams as diagrams: A categorification of the stability theorem. In *Topological Data Analysis*, pages 67–96. Springer International Publishing, 2020. doi:10.1007/978-3-030-43408-3_3.

- 7 Ulrich Bauer, Elizabeth Munch, and Yusu Wang. Strong equivalence of the interleaving and functional distortion metrics for Reeb graphs. In *31st International Symposium on Computational Geometry (SoCG 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPICs.SOCG.2015.461.
- 8 Ulrich J Bauer, Håvard Bakke Bjerkevik, and Benedikt Fluhr. Quasi-universality of Reeb graph distances. *arXiv preprint*, 2021. arXiv:1705.01690.
- 9 Håvard Bakke Bjerkevik and Michael Lesnick. ℓ^p -distances on multiparameter persistence modules. *arXiv preprint*, 2021. arXiv:2106.13589.
- 10 Andrew J Blumberg and Michael Lesnick. Universality of the homotopy interleaving distance. *arXiv preprint*, 2017. arXiv:1705.01690.
- 11 Brian Bollen, Erin W. Chambers, Joshua A. Levine, and Elizabeth Munch. Reeb graph metrics from the ground up. *arXiv preprint*, 2021. arXiv:2110.05631.
- 12 Peter Bubenik, Vin de Silva, and Jonathan Scott. Metrics for generalized persistence modules. *Foundations of Computational Mathematics*, 15(6):1501–1531, 2015. doi:10.1007/s10208-014-9229-5.
- 13 Peter Bubenik, Jonathan Scott, and Donald Stanley. Exact weights, path metrics, and algebraic Wasserstein distances. *arXiv preprint*, 2018. arXiv:1809.09654.
- 14 Peter Bubenik and Jonathan A Scott. Categorification of persistent homology. *Discrete & Computational Geometry*, 51(3):600–627, 2014. doi:10.1007/s00454-014-9573-x.
- 15 Gabriel Cardona, Arnau Mir, Francesc Rosselló, Lucia Rotger, and David Sánchez. Cophenetic metrics for phylogenetic trees, after Sokal and Rohlf. *BMC bioinformatics*, 14(1):1–13, 2013. doi:10.1186/1471-2105-14-3.
- 16 Robert Cardona, Justin Curry, Tung Lam, and Michael Lesnick. The universal ℓ^p -metric on merge trees. *arXiv preprint*, 2021. arXiv:2112.12165.
- 17 Gunnar Carlsson and Facundo Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, 11(47):1425–1470, 2010. URL: <http://jmlr.org/papers/v11/carlsson10a.html>.
- 18 Hamish Carr, Jack Snoeyink, and Ulrike Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003. doi:10.1016/S0925-7721(02)00093-7.
- 19 Hamish Carr, Jack Snoeyink, and Michiel van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *IEEE Visualization 2004*, pages 497–504, 2004. doi:10.1109/VISUAL.2004.96.
- 20 Erin Wolf Chambers, Elizabeth Munch, and Tim Ophelders. A family of metrics from the truncated smoothing of Reeb graphs. In *37th International Symposium on Computational Geometry*, 2021. doi:10.4230/LIPICs.SoCG.2021.22.
- 21 Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J Guibas, and Steve Y Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 237–246, 2009. doi:10.1145/1542362.1542407.
- 22 Yen-Chi Chen. Generalized cluster trees and singular measures. *The Annals of Statistics*, 47(4):2174–2203, 2019. doi:10.1214/18-AOS1744.
- 23 David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have L^p -stable persistence. *Foundations of computational mathematics*, 10(2):127–139, 2010. doi:10.1007/s10208-010-9060-6.
- 24 William Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules. *Journal of Algebra and its Applications*, 14(05):1550066, 2015. doi:10.1142/S0219498815500668.
- 25 Justin Curry. The fiber of the persistence map for functions on the interval. *Journal of Applied and Computational Topology*, 2(3):301–321, 2018. doi:10.1007/s41468-019-00024-z.
- 26 Justin Curry, Haibin Hang, Washington Mio, Tom Needham, and Osman Berat Okutan. Decorated merge trees for persistent topology. *To Appear in the Journal of Applied and Computational Topology*, 2022. arXiv:2103.15804.

- 27 Justin Michael Curry. *Sheaves, Cosheaves and Applications*. PhD thesis, University of Pennsylvania, 2014.
- 28 Vin de Silva, Elizabeth Munch, and Amit Patel. Categorified Reeb graphs. *Discrete & Computational Geometry*, 55(4):854–906, 2016. doi:10.1007/s00454-016-9763-9.
- 29 Vin de Silva, Elizabeth Munch, and Anastasios Stefanou. Theory of interleavings on categories with a flow. *Theory and Applications of Categories*, 33(21):583–607, 2018. URL: <http://www.tac.mta.ca/tac/volumes/33/21/33-21.pdf>.
- 30 Barbara Di Fabio and Claudia Landi. The edit distance for Reeb graphs of surfaces. *Discrete & Computational Geometry*, 55(2):423–461, 2016. doi:10.1007/s00454-016-9758-6.
- 31 Thomas Duquesne and Jean-François Le Gall. *Random trees, Lévy processes and spatial branching processes*. Number 281 in Astérisque. Société mathématique de France, 2002. URL: http://www.numdam.org/item/AST_2002__281__R1_0/.
- 32 Elena Farahbakhsh Touli. Fréchet-like distances between two rooted trees. *Journal of Algorithms and Computation*, 53(1):1–12, 2021. doi:10.22059/JAC.2021.81145.
- 33 Elena Farahbakhsh Touli and Yusu Wang. FPT-algorithms for computing Gromov-Hausdorff and interleaving distances between trees. In *European Symposium on Algorithms*, 2019. doi:10.4230/LIPIcs.ESA.2019.83.
- 34 Christoph Flamm, Ivo L. Hofacker, Peter F. Stadler, and Michael T. Wolfinger. Barrier trees of degenerate landscapes. *Zeitschrift für Physikalische Chemie*, 2002. doi:10.1524/zpch.2002.216.2.155.
- 35 Ellen Gasparovic, Elizabeth Munch, Steve Oudot, Katharine Turner, Bei Wang, and Yusu Wang. Intrinsic interleaving distance for merge trees. *arXiv preprint*, July 2019. arXiv:1908.00063.
- 36 Christopher Gold and Sean Cormack. Spatially ordered networks and topographic reconstructions. *International Journal of Geographical Information Systems*, 1(2):137–148, 1987. doi:10.1080/02693798708927800.
- 37 John A. Hartigan. Consistency of single linkage for high-density clusters. *Journal of the American Statistical Association*, 76(374):388–394, 1981. doi:10.1080/01621459.1981.10477658.
- 38 Masaki Kashiwara and Pierre Schapira. Persistent homology and microlocal sheaf theory. *Journal of Applied and Computational Topology*, 2(1):83–113, 2018. doi:10.1007/s41468-018-0019-z.
- 39 In So Kweon and Takeo Kanade. Extracting topographic terrain features from elevation maps. *CVGIP: Image Understanding*, 59(2):171–182, 1994. doi:10.1006/ciun.1994.1011.
- 40 Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015. doi:10.1007/s10208-015-9255-y.
- 41 Dmitriy Morozov, Kenes Beketayev, and Gunther Weber. Interleaving distance between merge trees. *Proceedings of Topology-Based Methods in Visualization*, 2013.
- 42 Elizabeth Munch and Anastasios Stefanou. The l^∞ -cophenetic metric for phylogenetic trees as an interleaving distance. In *Research in Data Science*, pages 109–127. Springer International Publishing, 2019. doi:10.1007/978-3-030-11566-1_5.
- 43 Amit Patel. Generalized persistence diagrams. *Journal of Applied and Computational Topology*, 1(3):397–419, 2018. doi:10.1007/s41468-018-0012-6.
- 44 Daniel Perez. On C^0 -persistent homology and trees. *arXiv preprint*, 2020. arXiv:2012.02634.
- 45 Daniel Perez. On the persistent homology of almost surely C^0 stochastic processes. *arXiv preprint*, 2020. arXiv:2012.09459.
- 46 Mathieu Pont, Jules Vidal, Julie Delon, and Julien Tierny. Wasserstein distances, geodesics and barycenters of merge trees. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):291–301, 2021. doi:10.1109/TVCG.2021.3114839.
- 47 Andrew Robinson and Katharine Turner. Hypothesis testing for topological data analysis. *Journal of Applied and Computational Topology*, 1(2):241–261, December 2017. doi:10.1007/s41468-017-0008-7.

- 48 Luis N Scoccola. *Locally persistent categories and metric properties of interleaving distances*. PhD thesis, The University of Western Ontario, 2020.
- 49 Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *arXiv preprint*, 2021. [arXiv:2006.16824](https://arxiv.org/abs/2006.16824).
- 50 Raghavendra Sridharamurthy, Talha Bin Masood, Adhitya Kamakshidasan, and Vijay Natarajan. Edit distance between merge trees. *IEEE Transactions on Visualization and Computer Graphics*, 26(3):1518–1531, 2020. [doi:10.1109/TVCG.2018.2873612](https://doi.org/10.1109/TVCG.2018.2873612).
- 51 Raghavendra Sridharamurthy and Vijay Natarajan. Comparative analysis of merge trees using local tree edit distance. *IEEE Transactions on Visualization and Computer Graphics*, 2021.
- 52 Katharine Turner. Medians of populations of persistence diagrams. *Homology, Homotopy and Applications*, 22(1):255–282, 2020. [doi:10.4310/HHA.2020.v22.n1.a15](https://doi.org/10.4310/HHA.2020.v22.n1.a15).

On Complexity of Computing Bottleneck and Lexicographic Optimal Cycles in a Homology Class

Erin Wolf Chambers  

Saint Louis University, MO, USA

Salman Parsa 

University of Utah, Salt Lake City, UT, USA

Hannah Schreiber  

Saint Louis University, MO, USA

Abstract

Homology features of spaces which appear in applications, for instance 3D meshes, are among the most important topological properties of these objects. Given a non-trivial cycle in a homology class, we consider the problem of computing a representative in that homology class which is optimal. We study two measures of optimality, namely, the lexicographic order of cycles (the lex-optimal cycle) and the bottleneck norm (a bottleneck-optimal cycle). We give a simple algorithm for computing the lex-optimal cycle for a 1-homology class in a closed orientable surface. In contrast to this, our main result is that, in the case of 3-manifolds of size n^2 in the Euclidean 3-space, the problem of finding a bottleneck optimal cycle cannot be solved more efficiently than solving a system of linear equations with an $n \times n$ sparse matrix. From this reduction, we deduce several hardness results. Most notably, we show that for 3-manifolds given as a subset of the 3-space of size n^2 , persistent homology computations are at least as hard as rank computation (for sparse matrices) while ordinary homology computations can be done in $O(n^2 \log n)$ time. This is the first such distinction between these two computations. Moreover, it follows that the same disparity exists between the height persistent homology computation and general sub-level set persistent homology computation for simplicial complexes in the 3-space.

2012 ACM Subject Classification Mathematics of computing \rightarrow Algebraic topology; Mathematics of computing \rightarrow Geometric topology; Theory of computation \rightarrow Computational geometry

Keywords and phrases computational topology, bottleneck optimal cycles, homology

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.25

Related Version *Full Version*: <https://arxiv.org/abs/2112.02380>

Funding *Erin Wolf Chambers*: This author was funded in part by the National Science Foundation through grants CCF-1614562, CCF-1907612, CCF-2106672, and DBI-1759807.

Salman Parsa: This author was funded in part by the Saint Louis University Research Institute and by NSF grant CCF-1614562.

Hannah Schreiber: This author was funded in part by the National Science Foundation through grant DBI-1759807.

1 Introduction

Topological features of a space are those features that remain invariant under continuous, invertible deformations of the space. Homology groups are one of the most important topological features which, while not a complete invariant of shape, nevertheless are computationally feasible and capture important structure, in the following sense. Let \mathbb{K} denote our space, which we will assume is a simplicial complex. For any dimension d , there is a homology



© Erin Wolf Chambers, Salman Parsa, and Hannah Schreiber;
licensed under Creative Commons License CC-BY 4.0

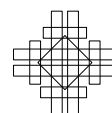
38th International Symposium on Computational Geometry (SoCG 2022).

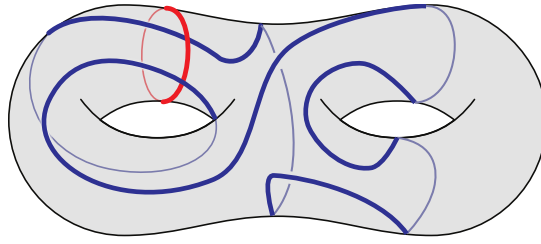
Editors: Xavier Goaoc and Michael Kerber; Article No. 25; pp. 25:1–25:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** The blue cycle is homologous (with \mathbb{Z}_2 coefficients) to the red cycle.

group¹ $H_d(\mathbb{K})$ that captures the d -dimensional structure present. The zero dimensional group encodes the connected components of \mathbb{K} ; the group $H_1(\mathbb{K})$ contains information about the closed curves in \mathbb{K} which can not be “filled” in the space (often described as handles); and the group $H_2(\mathbb{K})$ captures the voids in the space that could not be filled, etc.² For example, a hollow torus contains a single void and two classes of curves that are not “filled” in the space, and these features remain under continuous, invertible deformations of the shape.

Although the above intuitive description of the homology features is useful in many applications, in general, homology groups are algebraic objects defined for a simplicial complex (or a topological space) which do not easily translate to a canonical geometric feature. An element of a d -dimensional homology group is a homology class, and a homology class by definition contains a set of d -cycles, where cycles which are in the same class are called homologous to each other. Assume our complex \mathbb{K} is a 3D mesh and $d = 1$. A cycle under homology is a set of edges in the mesh, such that each vertex is incident to an even number of edges. A fixed cycle therefore corresponds to a fixed geometric feature of the mesh, while the homology class contains a large collection of these cycles. Cycles in the same class could be very different geometrically, see figure 1. Consequently, the knowledge of homology groups or Betti numbers (which are the dimensions of the homology groups) does not directly provide us with geometric features that lend themselves to interpretations that are necessary for many applications, especially in topological data analysis. Therefore, it is desirable to assign unique cycles, or those with known geometric features, to a homology class in a natural way. Much recent work has sought to define measures or weights on the cycles and then represent each homology class by some (ideally unique) cycle which optimizes that measure. This problem has been well-studied in the literature with many different measures proposed; see Section 1.2 for an overview of relevant results. Interestingly, sometimes optimizing the cycle is NP-hard and sometimes polynomial-time, depending on the measure, the classes of spaces one allows in the input, and the type of homology calculation. One of the more widely studied versions gives each edge a weight and then seeks the representative with minimum total length in the homology class. This problem is known as *homology localization*, and even its complexity varies widely depending on the space and the type of homology calculation. There is also a rich body of work that seeks to compute optimal cycles in persistent homology classes; again, we refer to Section 1.2 for details and citations.

In most of the paper we fix a simplicial complex \mathbb{K} , a fixed d , and a weight function given on d -simplices of \mathbb{K} . However, unlike traditional homology localization, for most of our work, it suffices to think of the weight function as an ordering of the simplices. We then consider

¹ In this work, we will always use \mathbb{Z}_2 coefficients, so that the homology groups are also vector spaces.

² Note that this is a high level, intuitive description; we refer the reader to [27, 24] for more precise definitions.

two measures that this ordering induces on the set of d -cycles. The first, already defined and studied in Cohen-Steiner et al. [11], is the lexicographic ordering on the chains. The second is a minmax measure we call the bottleneck norm, which assigns to each chain the maximum weight of a simplex in it. We note that computing the lexicographic-optimal cycle is at least as hard as computing a bottleneck-optimal cycle in a given homology class, as the lexicographic-optimal cycle is always bottleneck-optimal (but the reverse is not always true). In the rest of the paper, we often shorten lexicographical-optimal to lex-optimal.

1.1 Contributions

It is proved in [11] that the persistent homology boundary matrix reduction can be used to compute the lex-optimal cycle in any given homology class in cubic time in the size of the complex, for any dimension. In this paper, we begin by presenting a new simple algorithm that, given a (closed orientable) surface and a 1-dimensional cycle, computes a lex-optimal cycle homologous to the input cycle in $O(m \log m)$ time, where m is the size of a triangulation of the surface. We note that an algorithm with slightly better running-time ($O(n\alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function) is also given in [11] although their algorithm only works for cycles which are homologous to a boundary and satisfy some other restrictions, see [11, Problem 17].

The simplest setting after surfaces is perhaps 3-manifolds embedded in Euclidean 3-space, for instance solid 3D meshes. For simplifying run-time comparisons, we denote the sizes of the complexes in \mathbb{R}^3 by n^2 . Our main contribution in this paper is that given a system $Ax = b$ of linear equations with A a sparse³ 0-1 matrix, it is possible to construct in $O(n^2 \log n)$ time a 3-manifold embedded in \mathbb{R}^3 of size $O(n^2)$ such that solving the system for a solution x is equivalent to computing a bottleneck-optimal cycle in a given homology class. Our reductions remain true for integer homology and other fields \mathbb{Z}_p (with an appropriate definition of optimal cycles), albeit with slight changes in the run-time of the reductions.

In [13], Dey presents an algorithm for computing the persistent diagram of a height function for a complex in \mathbb{R}^3 (of size n^2) in $O(n^2 \log n)$ time. In addition, in the same running time a set of generators can be computed. From our reduction, it follows that, if the given function on the complex (which is a mesh in \mathbb{R}^3) is not a height function, then these computations cannot be done faster than rank computation for a sparse 0-1 matrix. This gives a first answer to the main question asked in [13], asking if efficient algorithms exist for the non-height functions. In other words, our results show that there is a disparity between the efficiency of algorithms for computing sub-level-set persistence for 3D meshes of height and of general functions.

Ordinary Betti numbers for complexes in \mathbb{R}^3 (of size n^2) can be computed in $O(n^2 \log n)$ time [12] (if a triangulation of the complement is also given). It follows from our reduction that computing persistence Betti numbers for an arbitrary function for complexes in \mathbb{R}^3 is as hard as computing the rank of a sparse 0-1 $n \times n$ matrix (even if a triangulation of the complement is given). To our knowledge, this is the first such distinction between persistent and ordinary homology computations.

We should also mention that the significance of the reductions, like the ones presented in Section 4, is not giving a lower bound for the problem in the complexity theoretic sense, as we have not done this, since we do not know if solving a sparse system has a non-trivial lower bound. Rather, the reductions show that the geometry of the problem does not help

³ By a sparse $n \times n$ matrix we mean that the number of non-zeros is at most cn for some constant c .

in improving trivial deterministic algorithms. For instance, one of our theorems tells a researcher of geometric methods that it is futile to try to find a deterministic $O(m \log m)$ algorithm that computes persistent Betti numbers for meshes in 3D if that researcher is not interested in improving the best run-time for matrix rank computation for sparse matrices. As mentioned before, an $O(m \log(m))$ algorithm exists if we are interested only in height functions. Here, m is size of the input mesh.

1.2 Related work

The Optimal Homologous Chain Problem (OHCP) is a well studied problem in computational topology, which specifies a particular cycle or homology class and asks for the “optimal” cycle in the same homology class. Similarly, the problem of homology localization [30] specifies a topological “feature” (usually a homology class, such as a handle or void), and asks for a representative of that class. Such representatives can be used for simplification, mesh parametrization, surface mapping, and many other problems.

Of course, computability and practicality often depend on the exact definition of “optimal”, with a wide range of variants. One natural notion of optimal is simply to assume the input complex has weights on the simplices, and to compute the representative of minimum length, area, or volume (depending on the dimension). Here, length (or area or volume) of a chain is computed as a weighted summation of the weights of its simplices; it then remains to specify the coefficients used when computing these objects, since the choice of coefficient can greatly affect the results. The resulting trade-offs can be quite subtle and surprising. For example, minimum length homologous cycles with \mathbb{Z} coefficients in the homology class of highest dimension, if homology is torsion free, reduces to linear programming and hence is solvable in polynomial time [15, 7]. In contrast, with \mathbb{Z}_2 coefficients the problem is NP-hard to compute, even on 2-manifolds [6, 5]. In fact, homology localization is NP-hard to approximate for \mathbb{Z}_2 coefficients within a constant factor even when the Betti number is constant [9], and APX-Hard but fixed parameter tractable with respect to the size of the optimal cycle [2, 3]. When coefficients are over \mathbb{Z}_k , the problem becomes Unique Games Conjecture hard to approximate [22]. Homology localization has also been studied under the lens of parameterized complexity, where it is fixed parameter tractable in treewidth of the underlying complex [1].

There has been considerable followup work on different variants of homology localization. One major line of work focuses on persistent homology generators, which are often related to homology localization but seek generators in a filtration which realize a particular persistent homology class [8, 4, 25, 28, 13, 17, 16]; again, there is high variance on notions of optimality for these generators and on input assumptions, both of which affect complexity. More directly related to this paper, as noted in the introduction, lexicographic minimum cycles under some ordering on the simplices have also been studied [11].

Hardness of computing ordinary homology for complexes in Euclidean spaces is discussed in [19], where a reduction to rank computation of sparse matrices is presented; the results of this paper thus in a sense extend those of [19].

We note that there are randomized and probabilistic algorithms for sparse matrix operations in almost quadratic time [29, 10]. As a result, our reductions do not apply for these types of algorithms, since they take $O(n^2 \log(n))$ time for a matrix with $O(n)$ non-zeros. It is natural to ask for a reduction that is linear in the size of the input A ; indeed, this presents an interesting direction of future research.

2 Bottleneck and lex-optimal cycles

Let \mathbb{K} be a simplicial complex and $\mathbb{K}_d = \{\sigma_0, \dots, \sigma_m\}$ be the set of d -dimensional simplices of \mathbb{K} . A *weight function* w on \mathbb{K}_d is an arbitrary function $w : \mathbb{K}_d \rightarrow \mathbb{R}_{>0} = \{r \in \mathbb{R} \mid r > 0\}$. Thus w is defined on the generators of the chain group $C_d(\mathbb{K})$. For simplicity, we assume that w is injective, i.e., simplices have distinct weights. For our purposes, such a weight function is equivalent to one with co-domain $\mathbb{N} - \{0\}$, or a total ordering of the simplices. If the weight function is not injective, then the edges with the same weight have exactly the same potential to appear in the optimal cycle and adding some small perturbations to their weights to distinguish them will not affect the consistency of the end result.

We extend w to the function $b_w : C_d(\mathbb{K}) \rightarrow \mathbb{N}$ as follows: for a chain $c \in C_d(\mathbb{K})$ of the form $c = \sum_{i=0}^m t_i \sigma_i$, where, $\forall i, t_i \in \mathbb{Z}_2$, we set

$$b_w(c) = \begin{cases} \max_{\substack{0 \leq i \leq m \\ t_i=1}} \{w(\sigma_i)\} & \text{if } c \neq 0 \\ 0 & \text{if } c = 0. \end{cases}$$

In other words, if we view a chain $c \in C_d(\mathbb{K})$ as a set of simplices, b_w assigns to c the maximum weight of a simplex in c . We call b_w the *bottleneck norm* on $C_d(\mathbb{K})$.

By the *maximum simplex*, we mean the simplex with the largest weight in the chain.

Although $C_d(\mathbb{K})$ is a finite vector space, the function b_w has properties analogous to a norm. First, it is non-negative. Second, assume x and y are chains and σ_x and σ_y are their maximum simplices. The maximum simplex of $x+y$ has weight at most $\max\{w(\sigma_x), w(\sigma_y)\} \leq w(\sigma_x) + w(\sigma_y)$. Hence b_w satisfies the triangle inequality. And third, clearly if $b_w(c) = 0$ then $c = 0$.

One can also define a lexicographic ordering on the d -chains based on the given weight function w , see also [11]. For this purpose, we order the d -simplices such that $\sigma < \sigma'$ if and only if $w(\sigma) < w(\sigma')$. We assume that the subscript of the σ_i respects the order. Let $c = \sum t_j \sigma_j$ and $c' = \sum t'_j \sigma_j$. We define $c <_L c'$ if there exists an index j_0 such that for $j > j_0, t_j = 1$ if and only if $t'_j = 1$, and $t_{j_0} = 0, t'_{j_0} = 1$. We write $c \leq_L c'$ if $c <_L c'$ or $c = c'$.

2.1 Problem definitions

In this section, we give formal definitions for our two main problems, the *Bottleneck-Optimal Homologous Cycle Problem (Bottleneck-OHCP)* and the *Lexicographic-Optimal Homologous Cycle Problem (Lex-OHCP)* [11], as well as defining optimal bases for homology groups.

Bottleneck-OHCP. Given a weight function w on \mathbb{K}_d , and a cycle $\zeta \in Z_d(\mathbb{K})$, compute a cycle z_* such that $[z_*] = [\zeta]$ and such that z_* minimizes the bottleneck norm. More formally, find z_* such that $b_w(z_*) = \min\{b_w(z) \mid z \in Z_d(\mathbb{K}), \exists c \in C_{d+1}(\mathbb{K}), z + \zeta = \partial c\}$.

In other words, the weight of the maximum simplex in z_* is minimized in the homology class of ζ . Therefore, we can also define the bottleneck weight function $b_w^* : H_d(\mathbb{K}) \rightarrow \mathbb{R}_{\geq 0}$ on the homology classes by using the minimum $[\zeta] \mapsto b_w(z_*)$. Thus the problem can also be formulated as computing the cycle which achieves $b_w^*(h)$ given any representative of the homology class h .

Lex-OHCP. Given a weight function w on \mathbb{K}_d , and a cycle $\zeta \in Z_d(\mathbb{K})$, compute the cycle z_* such that $[z_*] = [\zeta]$ and for any d -cycle y , if $[y] = [\zeta]$ then $z_* \leq_L y$.

We note that by our convention on the weight function, the lex-optimal cycle is always unique. Moreover, the lex-optimal cycle is also bottleneck-optimal, however, the converse is not true. Our reductions and hardness results are formulated for the bottleneck norm. Counter-intuitively, considering this intermediate problem simplifies our reductions and hardness proofs.

Optimal basis. For any suitable measure or weight function on the cycles we can define the corresponding optimal basis. Let \leq_p be some pre-order on the set of d -cycles $Z_d(\mathbb{K})$ such that every subset $A \subset Z_d(\mathbb{K})$ has some chain a , such that $\forall z \in A, a \leq_p z$.

With respect to this pre-order, we define the *optimal basis* for d -homology, as a set of cycles $B \subset Z_d(\mathbb{K})$, representing the homology classes generating $H_d(\mathbb{K})$, as follows. Put the smallest non-zero element of $Z_d(\mathbb{K})$ in B . Now, repeat the following until B is a representative basis for d -homology: let A be the union of the cycles in the classes that are not in the subspace generated by the classes represented in B . Put the smallest cycle of A in B .

In Section 3, we will describe a simple algorithm for computing the lex-optimal basis for the 1-dimensional homology of a surface.

2.2 The Sub-level bottleneck weight function

We defined the bottleneck weight function $b_w^* : H_d(\mathbb{K}) \rightarrow \mathbb{N}$ on homology classes using a weight function on d -simplices for some fixed dimension d . Here we give a second, more natural definition of a generalization of this weight function. Let $\omega : |\mathbb{K}| \rightarrow \mathbb{R}$ be a generic simplex-wise linear function. The sub-level set of a value $r \in \mathbb{R}$ is the set $|\mathbb{K}|_{\leq r} = \{x \in |\mathbb{K}| \mid \omega(x) \leq r\}$. For any d -cycle $\zeta \in Z_d(\mathbb{K})$, define $b_\omega(\zeta) := \min\{r \in \mathbb{R} \mid \exists z \in Z_d^s(\mathbb{K}_{\leq r}), \exists y \in C_{d+1}^s(\mathbb{K}), \zeta + z = \partial y\}$, where C_\bullet^s denotes the singular chain complex. Intuitively, $b_\omega(\zeta)$ is the smallest value of r such that a chain homologous to ζ in \mathbb{K} appears in the sub-level-set. This value of course depends only on the homology class of ζ . Thus, we have a weight function $b_\omega^* : H_d(\mathbb{K}) \rightarrow \mathbb{R}$.

► **Lemma 1.** *For any weight function w on d -simplices of \mathbb{K} , there is a generic simplex-wise linear function ω on the barycentric subdivision of \mathbb{K} , such that for any homology class $h \in H_d(\mathbb{K})$, $b_\omega^*(h') = b_w^*(h)$, where h' is the image of h in the subdivision.*

Proof. Let \mathbb{K}' denote the subdivision of \mathbb{K} . Recall that for each simplex σ of \mathbb{K} there is a vertex $v(\sigma)$ in \mathbb{K}' . If σ is a d -simplex, we set $\omega(v(\sigma)) = w(\sigma)$. For all other vertices v of \mathbb{K}' we define $b_\omega(v)$ to be a very small positive number. We then replace these weights with positive integers while maintaining their order. It is easy to check that our function satisfies the statement of the lemma. ◀

Note that we use the barycentric subdivision simply to give a finer level of granularity on the sub-level sets. This subdivision appears to be necessary for the construction of the function ω .

2.3 Bottleneck weight function and persistent homology

A homology class $h \in H(\mathbb{K})$ is a set of cycles such that the difference of any two of the cycles is a boundary chain. Homology classes are intuitively referred to as homological features. Persistent homology tries to measure the importance of these features. For details see [18].

Let the set of simplices of \mathbb{K} be ordered such that for each simplex σ , the simplices on the boundary of σ appear before σ in the ordering. For instance, this ordering can be given by the time that a simplex is added, if we are building the complex \mathbb{K} by adding a simplex at a time. Of course, we need the boundary of a simplex to be present before adding it. Let

$$\emptyset = \mathbb{K}_0 \subset \mathbb{K}_1 \cdots \subset \mathbb{K}_{n-1} \subset \mathbb{K}_n = \mathbb{K}$$

be the sequence of complexes such that \mathbb{K}_i consists of the first i simplices in the ordering. Such a sequence is called a *filtration*. For $j \geq i$, let $f^{i,j} : \mathbb{K}_i \subset \mathbb{K}_j$ be the inclusion and $f_{\#}^{i,j}$ the induced homomorphism on the chain groups. The homology groups $H_d(\mathbb{K}_i)$ change as we add simplices. We want to track homology features during these additions.

For $0 \leq i \leq j \leq n$, the d -dimensional *persistent homology group* $H_d^{i,j}$ is the quotient

$$H_d^{i,j} = \frac{f_{\#}^{i,j}(Z_d(\mathbb{K}_i))}{B_d(\mathbb{K}_j) \cap f_{\#}^{i,j}(Z_d(\mathbb{K}_i))}.$$

In words, this is the group of those homology classes of $H_d(\mathbb{K}_j)$ which contain cycles already existing in \mathbb{K}_i .

We give now an alternate description of the persistent homology classes. The cycles representing homology features allow us to relate the classes of different spaces to each other. We will consider, in each \mathbb{K}_i , a basis of homology and assign to each homology class in these bases a cycle which we call a *p-representative* cycle. Consider \mathbb{K}_i and let σ be a d -simplex such that $\mathbb{K}_i \cup \{\sigma\} = \mathbb{K}_{i+1}$. There are two possibilities for the change that adding σ causes in the homology groups of \mathbb{K}_i .

1. $[\partial_d(\sigma)] = 0$ in \mathbb{K}_i . This implies there is a d -chain b such that $\partial_d(b) = \partial_d(\sigma)$. Therefore, $\partial_d(b + \sigma) = 0$. It is easily seen that the cycle $z = b + \sigma$ is not a boundary in \mathbb{K}_{i+1} . We say that the cycle $b + \sigma$ and the class $h = [b + \sigma]$ are *born* at time $i + 1$ or at \mathbb{K}_{i+1} . It follows that $H_k(\mathbb{K}_{i+1}) = H_k(\mathbb{K}_i)$ for $k \neq d$ and $H_d(\mathbb{K}_{i+1}) = H_d(\mathbb{K}_i) \oplus ([z])$, where (x) means the \mathbb{Z}_2 -vector space generated by x . We take the cycle z to be the p-representative for the class h in \mathbb{K}_{i+1} . Moreover, If z' is a (inductively defined) p-representative for a homology class of \mathbb{K}_i we transfer it to be the p-representative of its class in \mathbb{K}_{i+1} .
2. $[\partial_d(\sigma)] \neq 0$ in \mathbb{K}_i . In this case, adding the simplex σ causes the class $h = [\partial(\sigma)]$ to become trivial. In other words, each $z \in [\partial(\sigma)]$ is now a boundary and this class is merged with the class 0. Since the p-representatives form a basis of homology, h can be written as a summation of these. The Elder Rule tells us that we declare that the youngest p-representative in this representation *dies* entering \mathbb{K}_{i+1} . Any other class still can be written as summation of existing p-representatives. Note that each p-representative now represents a possibly larger class.

For $0 \leq i \leq j \leq n$, the d -dimensional persistent homology group $H_d^{i,j}$ consists of the classes, in $H_d(\mathbb{K}_j)$, of those d -dimensional p-representatives which are born at or before \mathbb{K}_i . Therefore, the p-representatives persist through the filtration. At any i , they form a basis of the homology groups of \mathbb{K}_i , and their lifetime can be depicted using *barcodes*. The *persistence diagram* encodes the birth and death indices of p-representatives. Note that the non-trivial homology classes of \mathbb{K} are born at some index but never die. From the above explanation the following can be observed. We omit the proof.

► **Proposition 2.** *Let $h \in H_d(\mathbb{K}_i)$ be a homology class and assume $h = \sum t_j [z_j]$ where $t_j \in \mathbb{Z}_2$ and the z_j are p-representatives. Then $\sum t_j z_j$ is a bottleneck optimal cycle for h (with respect to the ordering giving rise to the filtration).*

Notice that there is a choice of b in the first case of the case analysis above. In general, the p -representatives are not lex-optimal cycles. However, if we choose b to be lex-optimal the p -representatives form a lex-optimal basis. This set of basis elements can be computed using the persistent homology boundary matrix reduction algorithm [18], as shown in [11]. This algorithm runs in $O(m^2\ell)$ time where ℓ is the number of d -simplices and m is the number of $d + 1$ simplices. Also using this basis, a lex-optimal cycle can be computed in any given class in $O(\ell^2)$ time [11]. Of course, these algorithms also compute a bottleneck optimal cycle for any given homology class.

3 An efficient algorithm for 2-dimensional manifolds

In this section, we present a simple algorithm that, given a combinatorial 2-manifold \mathbb{K} , weights on the edges, and a 1-dimensional homology class, computes a lex-optimal representative cycle in the given class. For simplicity, we consider only orientable manifolds without boundary.

Our input \mathbb{K} is an edge-weighted, orientable combinatorial 2-manifold, therefore, $S := |\mathbb{K}|$ is an orientable surface without boundary. Let m be the complexity of \mathbb{K} . Let z be an input cycle on the 1-skeleton. Note that if we want an input cycle z^S in S and not in \mathbb{K} , i.e. the cycle is not on the 1-skeleton, then we can compute an homologous cycle z on the 1-skeleton with less than m edges in $O(\ell)$ time, where ℓ is the number of intersections of z^S with the edges of \mathbb{K} .

We first construct a minimum spanning tree T of the 1-skeleton of \mathbb{K} with respect to the given weights. Let G be the dual graph of the 1-skeleton of \mathbb{K} . The weight of an edge in G is equal to the weight of its corresponding dual in \mathbb{K} . Let Q be the maximum spanning co-tree of \mathbb{K} in G and let Q^* be the edges of \mathbb{K} whose duals are in Q . As shown in [20, Lemma 1] T and Q^* are disjoint. Let L be the edges that are not in T nor in Q^* , and recall that the triple (T, Q, L) determines a polygonal schema P of $4g$ sides for S , where g is the genus of the surface. See Figure 2 as example. This means that if we cut the surface at $T \cup L$ we obtain a disk D , and there is an identification map $g : D \rightarrow S$ which will “re-glue” the disk into a surface. Each edge of $T \cup L$ appears twice around the disk, and each edge of Q^* is a diagonal of this disk, connecting two vertices of the disk. The cutting of the edges of $T \cup L$ and computing the disk D can be done in linear time. The two vertices of the disk that the edges in Q^* connect can also be computed in linear time, using previous work on computing the minimal homotopic paths [14, 21].

During our algorithm, we maintain a data structure \mathcal{Z} which stores a circular list of elements of \mathbb{Z}_2 . The circular list contains a node for each boundary edge of the disk D . Note that any edge of $T \cup L$ corresponds to two edges on the boundary of D and thus two nodes of \mathcal{Z} .

Algorithm. We compute the lex-optimal cycle z^* in the homology class of the input cycle z as follows: We start with every node of \mathcal{Z} at value 0. Then, for every edge e in z in $T \cup L$, we set one of the two nodes corresponding to e to 1 and keep the other one at 0. Finally, for all remaining edges e in z , which therefore are in Q^* , let $a(e)$ be one vertex and $b(e)$ be the other vertex which e connects in D . We add 1 to any node whose corresponding edge of ∂D is between $a(e)$ and $b(e)$ in clockwise order. At the end, we define the cycle z^* to be the cycle consisting of edges whose two corresponding nodes in \mathcal{Z} sum to 1.

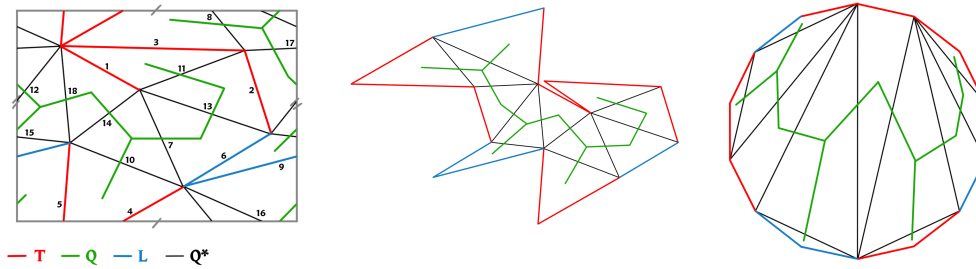


Figure 2 From left to right: a tree co-tree decomposition of a weighted, triangulated torus (with spanning tree shown in red, co-tree shown in green, and the set L in blue); the same torus cut along $T \cup L$; a redrawing of the resulting polygon.

Implementation of \mathcal{Z} . The data structure \mathcal{Z} has a single modifying operations: adding a value 1 to any node between two given nodes (inclusive) in clockwise order. In brief, to get a constant-per-operation run-time we accumulate the operations and update the data structure in a single pass. We give now more detail. \mathcal{Z} consists of an array A , whose cells are denoted by “nodes” to avoid any confusion with complex cells. Each node represents an edge of the boundary ∂D of D in the right order. For any edge e of z in Q^* , let $a'(e)$ be the first edge on the clockwise path between $a(e)$ and $b(e)$ in ∂D and $b'(e)$ the last edge. Additionally to a 0-1 value, each node c in A stores two values $s(c)$ and $f(c)$, where $s(c)$ resp. $f(c)$ is the number of edges e of z in Q^* whose $a'(e)$ resp. $b'(e)$ corresponds to c . For each e , the cost of updating these two numbers is constant. The final cycle can be computed by first computing the value of the first node $A[0]$ and then walking along A and updating the value as $A[i] = A[i - 1] + s(A[i]) - f(A[i - 1])$.

Correctness. Let $L = \{\ell_1, \dots, \ell_{2g}\}$, where the ℓ_i 's are sorted by increasing weight. Each edge ℓ_i defines a unique cycle when added to the tree T , let these cycles be denoted by $\Lambda = \{\lambda_1, \dots, \lambda_{2g}\}$. The following lemma is the key to our algorithm's correctness.

► **Lemma 3.** *Let $q \in Q^*$. Then there is a 1-chain $c <_L q$ in $T \cup L$, and a 2-chain d such that $\partial d = q + c$.*

Proof. The union of the edges in T and L form a cut graph G of the surface, in the sense that the closure of $S - |G|$ is a topological disk D . Every edge of $T \cup L$ appears twice on the boundary of D , and any $q \in Q^*$ is a diagonal in the polygon D . Let p_1 and p_2 be the two arcs such that $\partial D = p_1 \cup p_2$ and the endpoints of p_1 and p_2 coincide with those of q . Let $\tilde{p}_i \in C_1(\mathbb{K})$ be the 1-chain corresponding to p_i , $i = 1, 2$, i.e., $\tilde{p}_i = g_{\#}(p_i)$. Recall that $g_{\#}$ is the induced map on chain groups. Let d_1 be the 2-chain bounded by p_1 and q and let $\tilde{d}_1 = g_{\#}(d_1)$. We have $\partial \tilde{d}_1 = q + \tilde{p}_1$, where by q we denote this edge in D and S . We now claim that every edge in \tilde{p}_1 is smaller than q . Note that \tilde{p}_1 is a chain of $T \cup L$. We consider two cases. First, assume \tilde{p}_1 consists only of edges of T . In this case, it equals the unique path in T defined by the endpoints of q . Since T is a minimum spanning tree our claim is proved.

Second, assume that \tilde{p}_1 is not entirely in T . In this case we argue as follows. Let $\ell \in L$ and let ℓ_1 and ℓ_2 be the two copies of ℓ on ∂D . We claim that if ℓ_1 and ℓ_2 are on both of the arcs p_1 and p_2 (that is, if $\ell_1 \in p_1$ and $\ell_2 \in p_2$ or $\ell_2 \in p_1$ and $\ell_1 \in p_2$) then $\ell < q$. Assume for the sake of contradiction that $\ell > q$. Under these conditions, if we remove the dual of q

25:10 On Complexity of Computing Bottleneck and Lexicographic Optimal Cycles

from Q and add the dual of ℓ , we have reconnected the spanning co-tree split by removing q (since the effect of removing q from Q is adding it to L and thus cutting the disk D at q while the effect of adding ℓ to $Q - \{q\}$ is merging the resulting disks at ℓ_1 and ℓ_2 , thus again forming a single disk). Thus we have increased the weight of the spanning co-tree which is not possible. Therefore, $\ell < q$ or the two copies of ℓ appear on one of p_1 or p_2 . It follows that every edge of L (which appears once) in \tilde{p}_1 is smaller than q (since appearing twice cancels an edge). To finish the proof in this case, we claim that for every edge $t \in T \cap \tilde{p}_1$ there is an edge $\ell \in L \cap \tilde{p}_1$ such that $t < \ell$. It then follows that $\tilde{p}_1 < q$.

To prove the claim we argue as follows. $T \cup L$ is a graph on the 1-skeleton of K and it is standard and easy to show that any homology class $0 \neq h \in H_1(K)$ contains exactly one cycle of $T \cup L$. Let $z = q + x$ be the cycle formed by adding q to T , where x is the path on T . We have $z + \partial \tilde{d}_1 = z + \tilde{p}_1 + q = \tilde{p}_1 + x =: z'$ and z' is a non-empty cycle in $T \cup L$ (If z' were empty then $\tilde{p}_1 = x$, this is not possible since x is in T and \tilde{p}_1 is not in T by assumption). Thus z' can also be written as a non-empty summation of the λ_i . Each λ_i has the property that its unique edge $\ell_i \in L$ is larger than its edges in T . Since these ℓ_i are never cancelled, it follows that for each edge $t \in z' \cap T$ there is an edge $\ell \in z' \cap L$ such that $t < \ell$. Since the L -edges are in \tilde{p}_1 and not in x our claim is proved. \blacktriangleleft

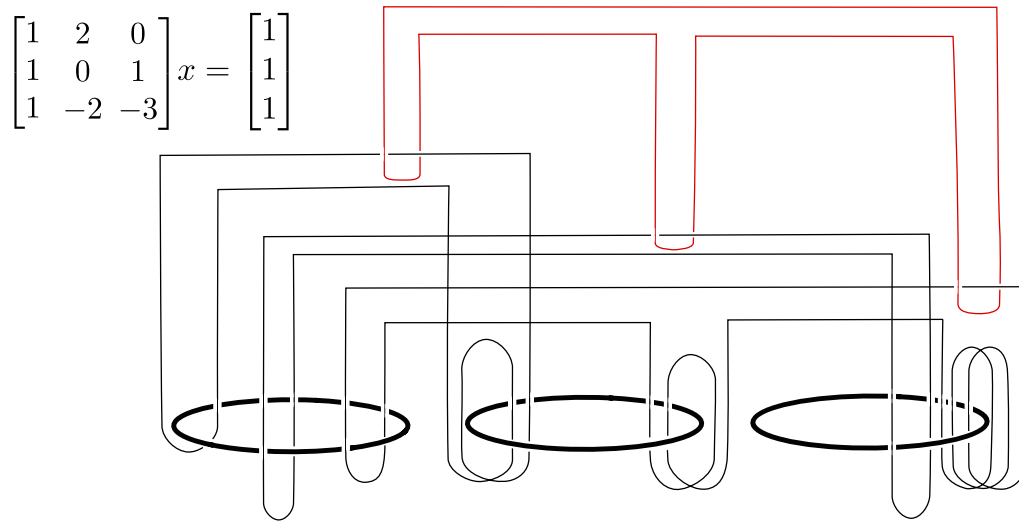
With some abuse of notation we also denote the chain on ∂D defined by the nodes of \mathcal{Z} with value 1 by \mathcal{Z} . Note that in the beginning of the algorithm $g_{\#}(\mathcal{Z}) = z$. The algorithm then repeatedly updates \mathcal{Z} by adding the chain p_1 returned by the above lemma to \mathcal{Z} and adding the chain $\tilde{p}_1 = g_{\#}(p_1)$ to z . It updates \mathcal{Z} such that at any time $g_{\#}(\mathcal{Z}) = z$. To finish the proof of correctness, it remains to show that the final cycle, namely the unique cycle z_* of $T \cup L$ in the class of z , is indeed the lex-min cycle. To see this, assume on the contrary that there is a cycle y such that $[y] = [z]$ and $y < z_*$. Since there is a unique cycle of any class in $T \cup L$, y has to contain an edge of Q^* hence can be made smaller, which contradicts minimality of y . Therefore, the cycles of $T \cup L$ are indeed the lex-min representatives of homology classes.

► Theorem 4. *Let \mathbb{K} be a simplicial complex which is a closed orientable combinatorial 2-manifold and let m be its number of simplices. There is an algorithm that computes a lex-optimal basis for the 1-dimensional homology of \mathbb{K} in $O(m \log(m))$ time. Moreover, we can compute a lex-optimal representative for any given 1-homology class within the same run-time.*

Proof. We have proved that the algorithm correctly computes the lex-optimal cycle homologous to z . We show that the basis Λ is lex-optimal basis. First note that by Lemma 3 every non-trivial cycle y is homologous to a cycle $y' \leq_L y$ such that y' is a subset of $T \cup L$. Since these must contain some ℓ_i , it follows that the smallest non-trivial cycle contains only ℓ_1 and edges of T , and hence is λ_1 .

Assume inductively that $\Lambda_i = \{\lambda_1, \dots, \lambda_i\}$ is a lex-optimal basis for the vector space $(\Lambda_i) \subset H_1(\mathbb{K})$. We claim that λ_{i+1} is the smallest cycle in classes in the set $H_1 - (\Lambda_i)$. Consider any non-trivial cycle y and decrease it to y' as above. To see that λ_{i+1} is the smallest cycle, note that $y' \cap L$ must contain some ℓ_j larger than λ_i , since otherwise $[y] \in (\Lambda_i)$; the smallest cycle with this property is λ_{i+1} .

Constructing T , the dual graph and Q takes at most $O(m \log m)$ time. Since we perform one update operation on \mathcal{Z} per edges of Q the total running time is $O(m \log m)$. \blacktriangleleft



■ **Figure 3** This figure is a link diagram for the reduction of the indicated linear system. The thick round circles are the λ_i , and the link components are drawn in thin black and red. For all of the crossings between \mathcal{L}_i , the vertical strand goes over the horizontal strand. In other words, these are not linked with each other. Note that in this example we are using integer matrix and integer linking number to showcase how the reduction works for integers. In the text we are concerned with 0-1 matrices. One also sees here how the need for generating large linking numbers (in absolute value) increases the complexity of the reduction.

4 Reductions

In this section, we first reduce solving a system of linear equation $Ax = b$, with A sparse, to computing the bottleneck-optimal homologous cycle problem for a 3-manifold given as a subset of the Euclidean 3-space. We then use this reduction to deduce hardness results for similar homological computations for 3-manifolds and 2-complexes in 3-space. Due to space constraints, some proofs of this section can only be found in the full version of the paper.

Let $A = (a_{ij})$, $i, j \in \{1, \dots, n\}$, be an $n \times n$ square matrix with values in \mathbb{Z}_2 . Let A_i denote the i -th column, and A_i^t denote the i -th row of A . Let $x = (x_1, \dots, x_n)$ be the vector of the n variables of the system $Ax = b$, and $b = (b_1, \dots, b_n)$.

From the given system $Ax = b$, we first construct a link diagram \mathcal{L}' . We start by drawing n round circles in the plane, whose collection we denote by $\Lambda' = \{\lambda'_1, \dots, \lambda'_n\}$; see the thick circles in the Figure 3 for an illustration. For each row A_i^t of A , we draw a component of the link \mathcal{L}' , denoted L'_i , such that its linking number is non-zero with λ'_j if and only if $a_{ij} = 1$; this can be accomplished simply by linking L'_i appropriately with λ'_i depending on the value of a_{ij} . As we wish the L'_i 's to not link with each other, any crossings between a fixed L'_i and L'_j are simply set to be all over (or all under), so that they will remain unlinked. Again, we refer to Figure 3, where example knots L'_1, L'_2 , and L'_3 are depicted by thin black lines. We add one final knot, which we denote as ζ' , to the link \mathcal{L}' so that its linking number with λ'_i is non-zero if and only if $b_i = 1$; this can be accomplished by linking ζ' once with each L'_i . See the top knot shown in red in Figure 3 for an illustration.

► **Lemma 5.** *Let A be such that each row of A has at most c non-zero entries. Then the link diagram \mathcal{L}' has $O(cn^2)$ crossings.*

In the next step, we construct a spatial link \mathcal{L} from the link diagram \mathcal{L}' , such that the knots appear in the 1-skeleton of a triangulation of a 3-ball. This is standard and can be done in $O(m \log(m))$ -time where m is the number of crossings of the link diagram \mathcal{L}' [23,

25:12 On Complexity of Computing Bottleneck and Lexicographic Optimal Cycles

Lemma 7.1]. The resulted space has complexity $O(m)$. Our diagram has $O(n^2)$ many crossings, therefore this construction takes $O(n^2 \log n)$ time and we obtain triangulation of a ball with complexity $O(n^2)$. The spatial link \mathcal{L} corresponding to \mathcal{L}' is a set of disjoint simple closed curves in the 1-skeleton of a triangulation of a 3-ball B^3 . We denote the spatial knots corresponding to L'_i by L_i , and analogously we name other components of \mathcal{L} .

Consider the sub-link \mathcal{N} of \mathcal{L} consisting of the components L_i . We define the manifold M to be the link-complement of the link \mathcal{N} . This link-complement, by definition, is obtained by removing the interior of a thin polyhedral tubular neighborhood of each component of \mathcal{N} . This construction is again standard, and a triangulation of M can be constructed in linear time from the spatial link [23]. Therefore, the 3-manifold M is a subset of a 3-ball B^3 , and has n boundary components. By extra subdivisions, if necessary, we can make sure that in the interior of M , ζ , λ_i and b are simple, disjoint, closed curves in the 1-skeleton. To do this, it is enough to make sure this property holds in every tetrahedron.

The cycle ζ is the input cycle in our instance of the bottleneck-optimal homologous cycle problem. We still need to define our edge weights, which will be based on an ordering of the edges of M . Let $\{e_{ij}\}$ be the set of edges in the cycle λ_i . First, we make sure that every edge not in some λ_i is larger than any e_{ij} . Second, if $i < i'$, we make sure that, for all j, j' , e_{ij} is smaller than $e_{i'j'}$. This finishes construction of our problem instance.

Let μ_i , $i \in \{1, \dots, n\}$, be meridians of the knots L_i in M . This is a circle on the boundary component of M corresponding to L_i . It is well-known that the homology group $H_1(M)$ is a \mathbb{Z}_2 -vector space with the basis $\{[\mu_1], \dots, [\mu_n]\}$ isomorphic to \mathbb{Z}_2^n .

► **Lemma 6.** *The following hold:*

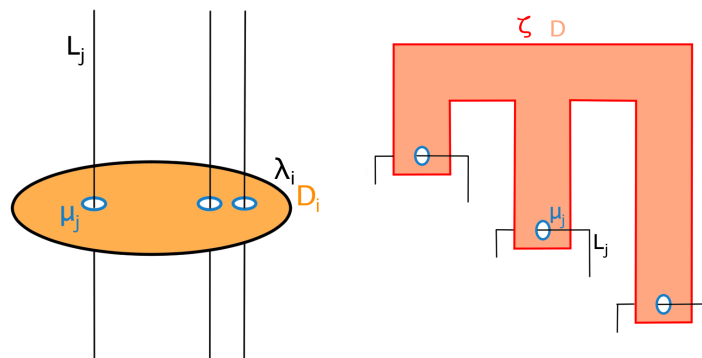
1. *If there is a vector $x \in \mathbb{Z}_2^n$ such that $Ax = b$ then any bottleneck-optimal cycle in the class $[\zeta]$ is a summation of the cycles λ_j .*
2. *If there exists a bottleneck-optimal cycle z_* in the class $[\zeta]$ such that $z_* = \sum_{i=1}^n x_i \lambda_i$ then the vector $x = (x_1, \dots, x_n)^t$ is a solution to $Ax = b$.*

Proof. First, observe that for the class $[\lambda_j]$ we have $[\lambda_i] = \sum_{j=1}^n a_{ji} [\mu_j]$. The left of Figure 4 depicts a 2-chain realizing this relation. If we map the basis element $[\mu_j]$ to the j -th standard basis element e_j , then we have defined an isomorphism $H_1(M) \cong \mathbb{Z}_2^n$ in which the class $[\lambda_i]$ maps to the column A_i . Second, note that, with a similar argument, $[\zeta] = \sum_{i=1}^n b_i [\mu_i]$, see Figure 4 right. Thus $[\zeta]$ maps to b under the isomorphism. It follows that $Ax = b$ if and only if $\sum x_j [\lambda_j] = [\zeta]$. The second statement follows.

If x is a solution to $Ax = b$, then the cycle $z = \sum x_j \lambda_j$ belongs to the class $[\zeta]$. Any cycle which is not entirely a subset of the edges of the λ_i 's, and hence a summation of the λ_i , contains some edge which is larger than all the edges of the λ_i 's and therefore has weight more than z . It follows that any bottleneck-optimal cycle is a summation of the λ_i or, a subset of them, since these are disjoint simple cycles. This proves the first statement. ◀

► **Theorem 7.** *Solving the system of equations $Ax = b$ where A is a sparse \mathbb{Z}_2 -matrix reduces in $O(n^2 \log n)$ time to the bottleneck-optimal homologous cycle problem with \mathbb{Z}_2 -coefficients for a 3-manifold of size $O(n^2)$ given as a subset of \mathbb{R}^3 .*

Proof. Given the system $Ax = b$ we have already constructed our instance. If the bottleneck-optimal cycle z_* returned by any algorithm that solves the Bottleneck-OHCP problem uses only edges in $\bigcup \lambda_i$, then the second statement of Lemma 6 implies that we can find a solution by determining which λ_i appear in z_* . This can be done in linear time. On the other hand, if z_* uses some edge not in $\bigcup \lambda_i$, then there is no solution to the system by the first statement of Lemma 6. ◀



■ **Figure 4** $\partial D_i = \lambda_i + \sum_j a_{ji} \mu_j$ (left), $\partial D = \zeta + \sum_j b_j \mu_j$ (right).

Although we have not defined the integer homology groups, it is almost immediate that the above reduction works also with \mathbb{Z} -coefficients.

► **Corollary 8.** *The (1-dimensional) lex-optimal homologous cycle problem for 3-manifolds in \mathbb{R}^3 of size n^2 cannot be solved more efficiently than the time required to solve a system of equations $Ax = b$ with A a sparse $n \times n$ matrix, if the latter time is $\Omega(n^2 \log(n))$.*

As noted before, the persistent boundary reduction algorithm can compute a lex-optimal cycle in $O(lm^2)$ time [11], where m is the number of $d + 1$ -simplices and l is the number of d -simplices. Although a set of persistent generators can be computed in matrix multiplication time [26], we do not know that the lex-optimal cycle can be found in matrix multiplication time, as it is unclear if the divide and conquer strategy from [26] would work on our problem.

► **Corollary 9.** *A set of sub-level-set persistent homology generators for a 3-manifold M or a 2-complex \mathbb{K} of size n^2 in \mathbb{R}^3 and a generic simplex-wise linear function $f : M \rightarrow \mathbb{R}$ cannot be computed more efficiently than the time required to compute a maximal set of independent columns in an $n \times n$ sparse matrix A , if the latter time is $\Omega(n^2 \log(n))$.*

As noted in the introduction, the above results are in a strong contrast with the results of Dey [13]. In other words, if the complex is of size $O(n^2)$ and the given function on the simplicial complex \mathbb{K} is a height function then one can compute the generators in $O(n^2 \log n)$ time, whereas, for a general function, one cannot do better than computing a maximal set of independent columns for a given sparse matrix A of size n . To the best of our knowledge, the best deterministic algorithm for this operation takes at least $O(n^\omega)$ time, where ω is the exponent of matrix multiplication.

► **Corollary 10.** *The persistence diagram for a 2-complex or a 3-manifold of size n^2 in \mathbb{R}^3 and a generic simplex-wise linear function $f : |\mathbb{K}| \rightarrow \mathbb{R}$ cannot be computed more efficiently than the time required to compute the rank of a sparse $n \times n$ matrix A , if the latter time is $\Omega(n^2 \log(n))$.*



Again the above theorem should be compared with results of [13], where the persistence is computed in $O(n^2 \log n)$ time for a 2-complex in 3-space of size n^2 and a height function.

References

- 1 Nello Blaser and Erlend Raa Vågset. Homology localization through the looking-glass of parameterized complexity theory, 2020. [arXiv:2011.14490](#).
- 2 Glencora Borradaile, William Maxwell, and Amir Nayyeri. Minimum bounded chains and minimum homologous chains in embedded simplicial complexes. In *Proceedings of the 36th International Symposium on Computational Geometry (SoCG 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- 3 Oleksiy Busaryev, Sergio Cabello, Chao Chen, Tamal K. Dey, and Yusu Wang. Annotating simplices with a homology basis and its applications. In Fedor V. Fomin and Petteri Kaski, editors, *Algorithm Theory – SWAT 2012*, pages 189–200, 2012.
- 4 Oleksiy Busaryev, Tamal K. Dey, and Yusu Wang. Tracking a generator by persistence. In My T. Thai and Sartaj Sahni, editors, *Computing and Combinatorics*, pages 278–287, 2010.
- 5 Erin W. Chambers, Jeff Erickson, Kyle Fox, and Amir Nayyeri. Minimum cuts in surface graphs, 2019. [arXiv:1910.04278](#).
- 6 Erin W. Chambers, Jeff Erickson, and Amir Nayyeri. Minimum cuts and shortest homologous cycles. In *Proceedings of the 25th International Symposium on Computational Geometry (SoCG 2009)*. ACM Press, 2009. doi:10.1145/1542362.1542426.
- 7 Erin W. Chambers and Mikael Vejdemo-Johansson. Computing minimum area homologies. *Computer Graphics Forum*, 34(6):13–21, November 2014. doi:10.1111/cgf.12514.
- 8 Chao Chen and Daniel Freedman. Measuring and computing natural generators for homology groups. *Computational Geometry*, 43(2):169–181, 2010. Special Issue on the 24th European Workshop on Computational Geometry (EuroCG’08). doi:10.1016/j.comgeo.2009.06.004.
- 9 Chao Chen and Daniel Freedman. Hardness results for homology localization. *Discrete & Computational Geometry*, 45(3):425–448, January 2011. doi:10.1007/s00454-010-9322-8.
- 10 Ho Yee Cheung, Tsz Chiu Kwok, and Lap Chi Lau. Fast matrix rank algorithms and applications. *Journal of the ACM*, 60(5), October 2013. doi:10.1145/2528404.
- 11 David Cohen-Steiner, André Lieutier, and Julien Vuillamy. Lexicographic Optimal Homologous Chains and Applications to Point Cloud Triangulations. In *36th International Symposium on Computational Geometry (SoCG 2020)*, pages 32:1–32:17. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.SoCG.2020.32.
- 12 Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 1995. doi:10.1016/0167-8396(95)00016-Y.
- 13 Tamal K. Dey. Computing height persistence and homology generators in \mathbb{R}^3 efficiently. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2019)*, pages 2649–2662, USA, 2019. Society for Industrial and Applied Mathematics.
- 14 Tamal K. Dey and Sumanta Guha. Transforming curves on surfaces. *Journal of Computer and System Sciences*, 58(2):297–325, 1999. doi:doi.org/10.1006/jcss.1998.1619.
- 15 Tamal K. Dey, Anil N. Hirani, and Bala Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. *SIAM Journal on Computing*, 40(4):1026–1044, January 2011. doi:10.1137/100800245.
- 16 Tamal K. Dey, Tao Hou, and Sayan Mandal. Persistent 1-cycles: Definition, computation, and its application, 2018. [arXiv:1810.04807](#).
- 17 Tamal K. Dey, Tao Hou, and Sayan Mandal. Computing minimal persistent cycles: Polynomial and hard cases. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020)*, pages 2587–2606, USA, 2020. Society for Industrial and Applied Mathematics.
- 18 Herbert Edelsbrunner and John Harer. *Computational Topology: an Introduction*. American Mathematical Society, 2010.
- 19 Herbert Edelsbrunner and Salman Parsa. On the computational complexity of betti numbers: Reductions from matrix rank. In *Proceedings of the 25th Annual ACM-SIAM Symposium*

- on *Discrete Algorithms (SODA 2014)*, pages 152–160. Society for Industrial and Applied Mathematics, 2014. doi:10.1137/1.9781611973402.11.
- 20 David Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2003)*, pages 599–608. Society for Industrial and Applied Mathematics, 2003.
 - 21 Jeff Erickson and Kim Whittlesey. Transforming curves on surfaces redux. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2013)*. Society for Industrial and Applied Mathematics, January 2013. doi:10.1137/1.9781611973105.118.
 - 22 Joshua A. Grochow and Jamie Tucker-Foltz. Computational topology and the unique games conjecture. In *Proceedings of 34th International Symposium on Computational Geometry (SoCG 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
 - 23 Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger. The computational complexity of knot and link problems. *Journal of the ACM (JACM)*, 46(2):185–211, 1999.
 - 24 A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002. URL: <https://pi.math.cornell.edu/~hatcher/AT/AT.pdf>.
 - 25 Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G. Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, June 2016. doi:10.1073/pnas.1520877113.
 - 26 Nikola Milosavljević, Dmitriy Morozov, and Primož Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the 27th International Symposium on Computational Geometry (SoCG 2011)*, pages 216–225, New York, NY, USA, 2011. ACM. doi:10.1145/1998196.1998229.
 - 27 James R. Munkres. *Elements of algebraic topology*. CRC press, 2018.
 - 28 Ipppei Obayashi. Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 2(4):508–534, January 2018. doi:10.1137/17m1159439.
 - 29 Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, 32(1):54–62, 1986. doi:10.1109/TIT.1986.1057137.
 - 30 Afra Zomorodian and Gunnar Carlsson. Localized homology. *Computational Geometry*, 41(3):126–148, November 2008. doi:10.1016/j.comgeo.2008.02.003.

Parameterized Algorithms for Upward Planarity

Steven Chaplick  

Maastricht University, The Netherlands

Emilio Di Giacomo  

Università degli Studi di Perugia, Italy

Fabrizio Frati  

Roma Tre University, Rome, Italy

Robert Ganian  

Technische Universität Wien, Austria

Chrysanthi N. Raftopoulou  

National Technical University of Athens, Greece

Kirill Simonov 

Technische Universität Wien, Austria

Abstract

We obtain new parameterized algorithms for the classical problem of determining whether a directed acyclic graph admits an upward planar drawing. Our results include a new fixed-parameter algorithm parameterized by the number of sources, an XP-algorithm parameterized by treewidth, and a fixed-parameter algorithm parameterized by treedepth. All three algorithms are obtained using a novel framework for the problem that combines SPQR tree-decompositions with parameterized techniques. Our approach unifies and pushes beyond previous tractability results for the problem on series-parallel digraphs, single-source digraphs and outerplanar digraphs.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Human-centered computing → Graph drawings

Keywords and phrases Upward planarity, parameterized algorithms, SPQR trees, treewidth, treedepth

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.26

Related Version *Full Version*: <https://arxiv.org/abs/2203.05364>

Funding *Emilio Di Giacomo*: MIUR, grant 20174LF3T8, Dip. Ing. – UNIPG, grants RICBA19FM and RICBA20EDG.

Fabrizio Frati: MIUR, grant 20174LF3T8.

Robert Ganian: Austrian Science Fund (FWF) Project Y1329.

Chrysanthi N. Raftopoulou: NTUA research program IIEBE 2020.

Kirill Simonov: Austrian Science Fund (FWF) Project P31336.

Acknowledgements The authors thank Fabrizio Montecchiani and Giuseppe Liotta for fruitful discussions on the topic of upward planarity. This research was initiated at Dagstuhl Seminar 21293: Parameterized Complexity in Graph Drawing [19].

1 Introduction

A digraph is called *upward planar* if it admits an upward planar drawing, that is, a planar drawing where all edges are oriented upward. The problem of upward planarity testing (UPWARD PLANARITY) and constructing an associated upward planar drawing arises, among others, in the context of visualization of hierarchical network structures; application domains include project management, visual languages and software engineering [2]. Upward planarity



© Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Robert Ganian, Chrysanthi N. Raftopoulou, and Kirill Simonov; licensed under Creative Commons License CC-BY 4.0

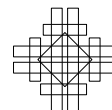
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 26; pp. 26:1–26:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



is the most prominent notion of planarity that is inherently directed, and also has classical connections to the theory of ordered sets: the orders arising from the transitive closure of upward planar single-source digraphs have bounded dimension [30].

Since the introduction of the notion, UPWARD PLANARITY has become the focus of extensive theoretical research. The problem has been shown to be NP-complete more than 25 years ago [20, 21], but the first polynomial-time algorithms for restricted variants of UPWARD PLANARITY have been published even earlier [25, 26]. Among others, the problem is known to be polynomial-time tractable when G is provided with a planar embedding [3] (which also implies polynomial-time tractability for triconnected DAGs, since these admit a single planar embedding), or when restricted to the class of outerplanar DAGs [28], DAGs whose underlying graph is series-parallel [13], and most prominently single-source DAGs [2, 5, 26].

In spite of the number of results on UPWARD PLANARITY that analyze the classical complexity of the problem on specific subclasses of instances, the problem was up to now mostly unexplored from the more fine-grained perspective of parameterized complexity analysis [10, 15]. In particular, while it was known that UPWARD PLANARITY is fixed-parameter tractable when parameterized by the cyclomatic number of the input DAG (or, equivalently, the feedback edge number of the underlying undirected graph) [7], by the number of triconnected components and cut vertices [23], or the number of triconnected components plus the maximum diameter of a split component [13], the complexity of the problem under classical structural parameterizations has remained completely open.

Contribution. We develop a novel algorithmic framework for solving UPWARD PLANARITY which combines parameterized dynamic programming with the SPQR-tree decompositions of planar graphs [12, 22, 24]. In essence, our framework uses a characterization of the “shapes” of faces in an upward planar drawing that is inspired by earlier work on the notion of spirality [3, 13] and reduces UPWARD PLANARITY to the task of handling the “rigid” nodes in these decompositions. Informally, the task that needs to be handled there can be stated as follows: what are all the possible ways to combine the possible shapes of the children of a rigid node to obtain an upward planar drawing for the node itself? The framework is formalized in the form of a general “Interface Lemma” (Lemma 13) which can be complemented with numerous parameterizations as well as other algorithmic approaches.

In the remainder of this article, we use this framework to push the boundaries of tractability for UPWARD PLANARITY. Our first result in this direction is a fixed-parameter algorithm for UPWARD PLANARITY parameterized by the number of sources in the input graph. This result generalizes the polynomial-time tractability of the single-source case [2, 5] and answers an open question from a recent Dagstuhl seminar [19]. On a high level, we use the Interface Lemma to reduce the problem to a case where almost all children of a rigid node have a simple shape, and we show how this can be handled via a flow network approach.

Having established the tractability of instances with few sources, we turn towards understanding which structural properties of the underlying undirected graph can be used to solve UPWARD PLANARITY efficiently. In this context, apart from the fixed-parameter tractability of UPWARD PLANARITY parameterized by the feedback edge number [7], nothing was known about whether the more widespread “decompositional” parameters can be used to solve the problem. The parameters that will be of interest here are *treewidth* [29], the most prominent structural graph parameter, and *treedepth* [27], the arguably best known parameter that lies below treewidth in the parameter hierarchy (see, e.g., [1, Figure 1]).

To obtain new boundaries of tractability for UPWARD PLANARITY with respect to these two parameters, we first show that the problem posed by the Interface Lemma can be restated as a purely combinatorial problem on a suitable combinatorization of the embedding

of the graph represented by the rigid node, and – crucially – that a bound on the input graph’s treewidth also implies a bound for the treewidth of this combinatorization. Once that is done, we design a non-trivial dynamic program that exploits this treewidth bound to handle the rigid nodes, which together with the Interface Lemma allows us to solve UPWARD PLANARITY. This yields an XP-algorithm for UPWARD PLANARITY parameterized by the treewidth of the underlying undirected graph – a result which unifies and generalizes the polynomial-time tractability of UPWARD PLANARITY on outerplanar as well as series-parallel graphs [13, 28]. Furthermore, a more detailed analysis of the dynamic program reveals that the same algorithm runs in fixed-parameter time when parameterized by treedepth.

Due to space limitations some proofs are omitted and can be found in [8].

2 Preliminaries

We refer to the usual sources for graph drawing and parameterized complexity terminology [10, 11, 14, 15]. We use $N_G(v)$ to denote the set of vertices adjacent to a vertex v in a graph G .

Upward planar drawings and embeddings. A *planar embedding* is an equivalence class of planar drawings of a graph, where two drawings are equivalent if the clockwise order of the edges incident to each vertex is the same and the outer faces are delimited by the same walk.

A vertex in a digraph is a *switch* if it is a source or a sink, and it is a *non-switch* otherwise. The *underlying graph* of a digraph is the undirected graph obtained from the digraph by ignoring the edge directions. A drawing of a digraph is *upward* if every edge is represented by a Jordan arc monotonically increasing from the source to the sink of the edge, and it is *upward planar* if it is both upward and planar. A digraph is *upward planar* if it admits an upward planar drawing; we use UPWARD PLANARITY to denote the problem of determining whether a digraph is upward planar; w.l.o.g., we assume that the input digraph is connected.

In an upward planar drawing Γ of a digraph G , an *angle* represents an incidence between a vertex v and a face f . The angle is either *flat* (if precisely one of the two edges incident to v and f is incoming at v), *large* (if v is a switch vertex and the angle has more than 180° in Γ), or *small* (otherwise) [3]; the latter two cases are jointly called *switch angles*. Then Γ defines an *angle assignment*, which assigns the value -1 , 0 , and 1 to each small, flat, and large angle, respectively, in every face of Γ . The angle assignment, together with the planar embedding of the underlying graph of G in Γ , constitutes an *upward planar embedding* of G .

The angle assignments that enhance a planar embedding into an upward planar embedding have been characterized by Didimo et al. [13], building on the work by Bertolazzi et al. [3]. Note that, once the planar embedding \mathcal{E} of a digraph G is specified, then so are the angles of the faces of \mathcal{E} ; in particular, whether an angle is flat or switch only depends on \mathcal{E} . Consider an angle assignment for \mathcal{E} . If v is a vertex of G , we denote by $n_i(v)$ the number of angles at v that are labeled i , with $i \in \{-1, 0, 1\}$. If f is a face of G , we denote by $n_i(f)$ the number of angles of f that are labeled i , with $i \in \{-1, 0, 1\}$. The cited characterization is as follows.

► **Theorem 1** ([3, 13]). *Let G be a digraph, \mathcal{E} be a planar embedding of the underlying graph of G , and λ be an assignment of each angle of each face in \mathcal{E} to a value in $\{-1, 0, 1\}$. Then \mathcal{E} and λ define an upward planar embedding of G if and only if the following properties hold:*

UP0 *If α is a switch angle, then $\lambda(\alpha) \in \{-1, 1\}$, and if α is a flat angle, then $\lambda(\alpha) = 0$.*

UP1 *If v is a switch vertex of G , then $n_1(v) = 1$, $n_{-1}(v) = \deg(v) - 1$, $n_0(v) = 0$.*

UP2 *If v is a non-switch vertex of G , then $n_1(v) = 0$, $n_{-1}(v) = \deg(v) - 2$, $n_0(v) = 2$.*

UP3 *If f is a face of G , then $n_1(f) = n_{-1}(f) - 2$ if f is an internal face and $n_1(f) = n_{-1}(f) + 2$ if f is the outer face.*

Treewidth and Treedepth. Here we consider the treewidth and treedepth of the underlying graphs¹. A *tree-decomposition* \mathcal{T} of a graph $G = (V, E)$ is a pair (T, χ) , where T is a tree (whose vertices we call *nodes*) rooted at a node r and χ is a function that assigns each node t a set $\chi(t) \subseteq V$ such that the following holds: for every $uv \in E$ there is a node t such that $u, v \in \chi(t)$, and for every vertex $v \in V$, the set of nodes t satisfying $v \in \chi(t)$ forms a nonempty subtree of T . The *width* of a tree-decomposition (T, χ) is the size of a largest set $\chi(t)$ minus 1, and the *treewidth* of the graph G , denoted $tw(G)$, is the minimum width of a tree-decomposition of G . The second structural parameter that we will be considering here is the *treedepth* of a graph G , denoted $td(G)$ [27]. A useful way of thinking about graphs of bounded treedepth is that they are (sparse) graphs with no long paths.

Expansion. In our algorithms, we will employ a linear-time preprocessing step called expansion to simplify the input digraphs so that every vertex has at most one incoming edge (in which case it is a *top* vertex) or at most one outgoing edge (in which case it is a *bottom* vertex) [2]. The expansion is obtained by replacing each non-switch vertex v with two new vertices v_1 and v_2 , which inherit the incoming and outgoing edges of v , respectively, and the edge (v_1, v_2) (called the *special edge* of v_1 and v_2). It is known that expansion preserves upward planarity, and it is possible to observe that it preserves biconnectivity, does not create new sources, and only increases treewidth and treedepth by at most a factor of 2.

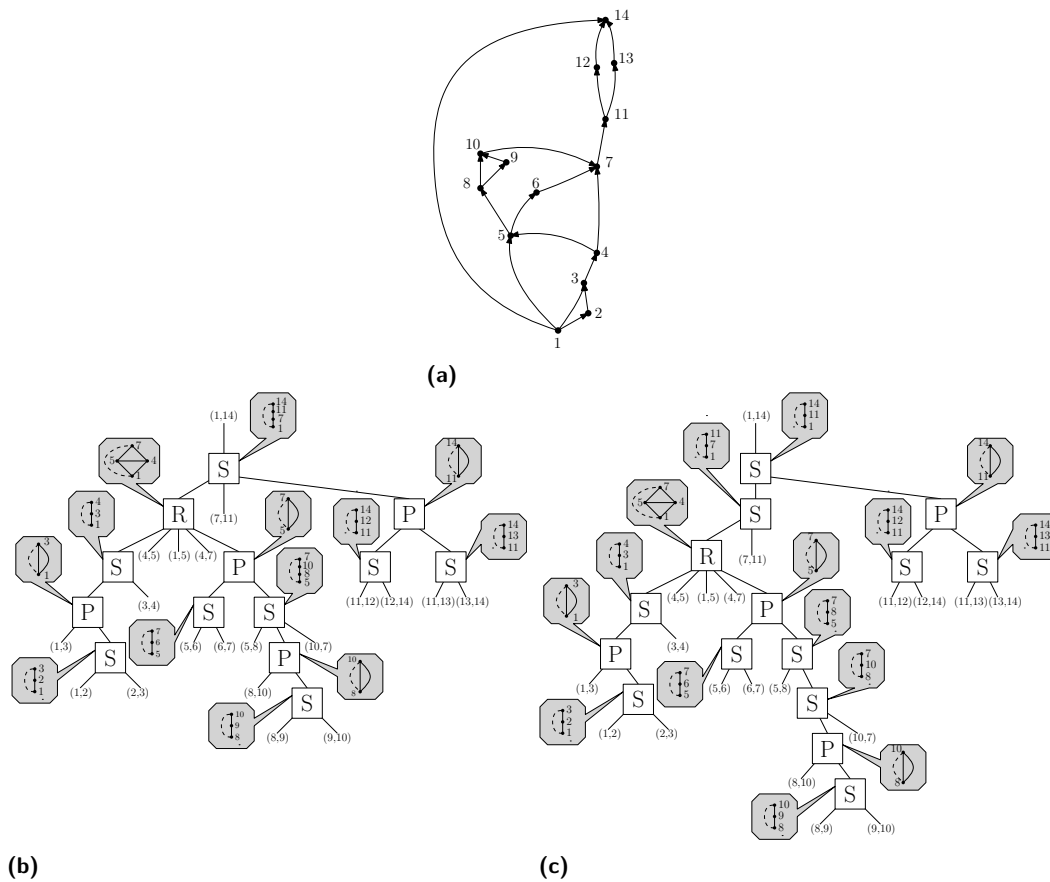
SPQR-tree decomposition. Let G be a biconnected undirected graph. A pair of vertices is a *separation pair* if its removal disconnects G . A *split pair* is either a separation pair or a pair of adjacent vertices. A *split component* of G with respect to a split pair $\{u, v\}$ is either an edge (u, v) or a maximal subgraph $G_{uv} \subset G$ such that $\{u, v\}$ is not a split pair of G_{uv} . A split pair $\{s', t'\}$ of G is *maximal* with respect to a split pair $\{s, t\}$ of G , if for every other split pair $\{s^*, t^*\}$ of G , there is a split component that includes the vertices s', t', s and t .

An *SPQR-tree* T of G with respect to an edge e^* is a rooted tree that describes a recursive decomposition of G induced by its split pairs [12]. Each node μ of T is associated with a split pair $\{u, v\}$ of G , where u and v are the *poles* of μ , with a subgraph G_μ of G , called the *pertinent graph* of μ , which consists of one or more split components of G with respect to $\{u, v\}$, and with a multigraph $sk(\mu)$, called the *skeleton* of μ , which represents the arrangement of such split components in G_μ . The edges of $sk(\mu)$ are called *virtual edges*. Each node μ of T whose pertinent graph is not a single edge has some children, each corresponding to a split components of G in G_μ . Each of these children is the root of a subtree of T . The nodes of T are of four types S, P, Q, and R. Q-nodes correspond to edges of G , while S-, P- and R-nodes correspond to so-called series, parallel and rigid compositions of the pertinent graphs of the children of the given node [12].

Note that each virtual edge e_i in the skeleton of a node μ of T *corresponds* to the pertinent graph G_{ν_i} of a child ν_i of μ . We say that G_{ν_i} is a *component* of G_μ . Figs. 1a and 1b show a planar graph and its SPQR-tree. To simplify our algorithms, we assume that every S-node of T has two children. If this is not the case, we can modify T to achieve this property (see Fig. 1c). An SPQR-tree T of an n -vertex planar graph has $\mathcal{O}(n)$ Q-, S-, P-, and R-nodes. Also, the total number of vertices of the skeletons for the nodes in T is $\mathcal{O}(n)$ [12].

When talking about an SPQR-tree T of a biconnected directed graph G , we mean an SPQR-tree of its underlying graph. Let μ be a node of T with poles u and v . A *uv -external upward planar embedding* of G_μ is an upward planar embedding of G_μ such that u and v

¹ Directed alternatives to treewidth exist, but are typically not well-suited for algorithmic applications [18].



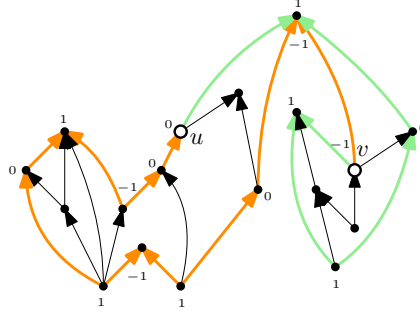
■ **Figure 1** (a) A planar DAG G . (b) An SPQR-tree of G . For each node that is not a Q-node, the skeleton is depicted together with a dashed edge to represent the rest of the graph; for each Q-node, the corresponding edge is shown. (c) An SPQR-tree of G whose S-nodes have exactly two children.

are incident to the outer face. In our algorithms, when testing the upward planarity of a digraph G , the fact that its SPQR-tree T is rooted at an edge e^* of G corresponds to the requirement that e^* is incident to the outer face of the upward planar embedding \mathcal{E} of G we are looking for. For each node μ of T , the restriction of \mathcal{E} to the vertices and edges of the pertinent graph G_μ of μ is a uv -external upward planar embedding of G_μ .

3 The Shapes of Components

Let G be a biconnected DAG, let T be an SPQR-tree of G rooted at an edge e^* , let μ be a node of T with poles u and v , and let \mathcal{E}_μ be a uv -external upward planar embedding of G_μ . Let λ be the angle assignment defined by \mathcal{E}_μ . The poles u and v identify two paths on the boundary of the outer face f_0 of \mathcal{E}_μ : the *left outer path* $P_l = \langle v_0 = u, v_1, \dots, v_k = v \rangle$ is the path that leaves f_0 on the left when walking from u to v ; the *right outer path* $P_r = \langle w_0 = u, w_1, \dots, w_h = v \rangle$ of \mathcal{E}_μ is the path that leaves f_0 on the right when walking from u to v ; see Fig. 2. For $i = 0, 1, \dots, k$, let α_i denote the angle at v_i inside f_0 and, for $i = 0, 1, \dots, h$, let β_i denote the angle at w_i inside f_0 . The *left-turn-number* $\tau_l(\mathcal{E}_\mu, u, v)$ of \mathcal{E}_μ is defined as $\sum_{i=1}^{k-1} \lambda(\alpha_i)$, while the *right-turn-number* $\tau_r(\mathcal{E}_\mu, u, v)$ of \mathcal{E}_μ is $\sum_{i=1}^{h-1} \lambda(\beta_i)$. Note that $\alpha_0 = \beta_0$ and $\alpha_k = \beta_h$ are the angles at u and v inside f_0 , respectively. The values $\lambda(\alpha_0)$ and $\lambda(\alpha_k)$ are also denoted by

$\lambda(\mathcal{E}_\mu, u)$ and $\lambda(\mathcal{E}_\mu, v)$, respectively. Finally, given a vertex $w \in \{u, v\}$, let $\rho_l(\mathcal{E}_\mu, w)$ denote the orientation of the edge e_l of P_l incident to w , that is, $\rho_l(\mathcal{E}_\mu, w) = in$ if e_l is an incoming edge for w , $\rho_l(\mathcal{E}_\mu, w) = out$ otherwise. Analogously, let $\rho_r(\mathcal{E}_\mu, w)$ denote the orientation of the edge e_r of P_r incident to w . The *shape description* of \mathcal{E}_μ is the tuple $\langle \tau_l(\mathcal{E}_\mu, u, v), \tau_r(\mathcal{E}_\mu, u, v), \lambda(\mathcal{E}_\mu, u), \lambda(\mathcal{E}_\mu, v), \rho_l(\mathcal{E}_\mu, u), \rho_r(\mathcal{E}_\mu, u), \rho_l(\mathcal{E}_\mu, v), \rho_r(\mathcal{E}_\mu, v) \rangle$; see Fig. 2.



■ **Figure 2** An upward planar embedding of a split component G_μ with poles u and v and shape description $\langle 3, 0, 0, -1, out, in, out, out \rangle$. The left (right) outer path is shown in green (orange).

There are some dependencies between the values of a shape description. For example, $\rho_l(\mathcal{E}_\mu, u) \neq \rho_r(\mathcal{E}_\mu, u)$ if $\lambda(\mathcal{E}_\mu, u) = 0$. As a further example, we have the following observation, which comes from Property **UP3** of Theorem 1 and uses the notation of this theorem.

▶ **Observation 2.** We have $\tau_l(\mathcal{E}_\mu, u, v) + \tau_r(\mathcal{E}_\mu, u, v) + \lambda(\mathcal{E}_\mu, u) + \lambda(\mathcal{E}_\mu, v) = 2$.

Recall that if u is a top or bottom vertex of G , then it has at most one incoming edge or at most one outgoing edge, respectively, which is called the *special edge* of u . If G_μ contains this edge, then G_μ is a *special component* for u , otherwise we say that G_μ is a *normal component* for u . Note that, if u is a source or a sink of G , then it has no special component.

▶ **Lemma 3.** We have $\tau_r(\mathcal{E}_\mu, u, v) = -\tau_l(\mathcal{E}_\mu, u, v) + h$, with $h \in \{0, 1, 2, 3, 4\}$.

4

 General Algorithm

Let G be an n -vertex biconnected expanded DAG whose underlying graph is planar and let T be an SPQR-tree of G . Let τ_{\min} and τ_{\max} be two integers with $\tau_{\min} \leq \tau_{\max}$ and let $\tau = \tau_{\max} - \tau_{\min} + 1$. We present a general algorithm to compute all possible shape descriptions of G with respect to T , and such that the left- and right-turn numbers of all shape descriptions for all pertinent graphs of T are in the range $[\tau_{\min}, \tau_{\max}]$. We visit the nodes of T bottom-up and we compute for each node μ its *feasible set* \mathcal{F}_μ , i.e., the set of all realizable shape descriptions of its pertinent graph G_μ . If $\mathcal{F}_\mu = \emptyset$, the process stops and G is not upward planar (under the above restrictions), otherwise we continue the traversal of T .

Storing feasible sets. For each node μ of T we associate a matrix $M(\mu)$ of size $(\tau_{\max} - \tau_{\min} + 1) \times 5$ where the element $M(\mu)[i, j]$ of the matrix contains all shape descriptions of G_μ with left turn-number $\tau_l = \tau_{\min} + i$ and right-turn-number $\tau_r = -\tau_l + j$. Note that by Lemma 3, τ_r can only take values in $[-\tau_l, -\tau_l + 4]$.

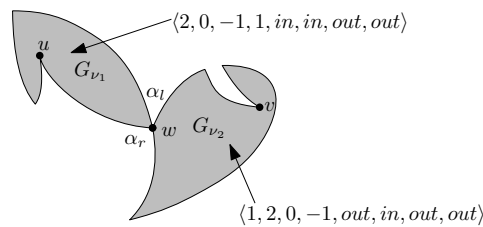
▶ **Lemma 4.** There are at most 18 shape descriptions with given left- or right-turn-number.

We describe how to compute the feasible set \mathcal{F}_μ of a node μ of T depending on its type.

Q-node. The pertinent graph G_μ is either the edge (u, v) or (v, u) . Hence, the feasible set consists of the tuple $\langle 0, 0, 1, 1, out, out, in, in \rangle$ or $\langle 0, 0, 1, 1, in, in, out, out \rangle$, respectively.

► **Lemma 5.** *Let μ be a Q-node of T . The feasible set \mathcal{F}_μ can be computed in $\mathcal{O}(1)$ time.*

S-node. Let ν_1 and ν_2 be the children of μ , with poles u, w and w, v , respectively. Let $\langle \tau_l^1, \tau_r^1, \lambda_u^1, \lambda_w^1, \rho_{l,u}^1, \rho_{r,u}^1, \rho_{l,w}^1, \rho_{r,w}^1 \rangle$ be a tuple in \mathcal{F}_{ν_1} and let $\langle \tau_l^2, \tau_r^2, \lambda_w^2, \lambda_v^2, \rho_{l,w}^2, \rho_{r,w}^2, \rho_{l,v}^2, \rho_{r,v}^2 \rangle$ be a tuple in \mathcal{F}_{ν_2} . Let α_l (resp. α_r) be the angle at w created by the two left (resp. right) outer paths of G_{ν_1} and G_{ν_2} (see Fig. 3). We assign the labels λ_l and λ_r to α_l and α_r respectively as follows: $\lambda_l = 0$ if $\rho_{l,w}^1 \neq \rho_{l,w}^2$ otherwise $\lambda_l \in \{-1, 1\}$, and $\lambda_r = 0$ if $\rho_{r,w}^1 \neq \rho_{r,w}^2$ otherwise $\lambda_r \in \{-1, 1\}$. Note that, by **UP1** it must be $\lambda_l + \lambda_r < 2$. For all possible values of λ_l and λ_r satisfying the previous constraints, we construct a candidate tuple $\langle \tau_l, \tau_r, \lambda_u, \lambda_v, \rho_{l,u}, \rho_{r,u}, \rho_{l,v}, \rho_{r,v} \rangle$ with: (i) $\tau_l = \tau_l^1 + \tau_l^2 + \lambda_l$, (ii) $\tau_r = \tau_r^1 + \tau_r^2 + \lambda_r$, (iii) $\lambda_u = \lambda_u^1$, (iv) $\lambda_v = \lambda_v^2$, (v) $\rho_{l,u} = \rho_{l,u}^1$, (vi) $\rho_{r,u} = \rho_{r,u}^1$, (vii) $\rho_{l,v} = \rho_{l,v}^2$, (viii) $\rho_{r,v} = \rho_{r,v}^2$. We accept the candidate tuple if and only if it satisfies Observation 2.



■ **Figure 3** Series composition. The resulting shape description is $\langle 2, 2, -1, -1, in, in, out, out \rangle$.

► **Lemma 6.** *Let μ be an S-node of T with children ν_1 and ν_2 . The feasible set \mathcal{F}_μ can be computed in $\mathcal{O}(\tau + |\mathcal{F}_{\nu_1}| \cdot |\mathcal{F}_{\nu_2}|)$ time.*

P-node. Let μ be a P-node with poles u and v and k children $\nu_1, \nu_2, \dots, \nu_k$. Let N' be a subset of the children of μ and let G'_μ be the subgraph of G_μ consisting of components $G_{\nu'}$ for $\nu' \in N'$. Consider a uv -external upward planar embedding \mathcal{E}'_μ of G'_μ . Denote by S' the sequence of shape descriptions of the components of G'_μ in the clockwise order in which they appear around u starting from the outer face. The sequence S' is the *shape sequence* of G'_μ with respect to \mathcal{E}'_μ . To describe S' we write: a^* (resp. a^+) to denote a subsequence of S' consisting of 0 (resp. 1) or more elements equal to a . We say that a shape description s' of G'_μ *corresponds* to S' if there exists an upward planar embedding of G'_μ with shape description s' and whose shape sequence is S' . Let S be a sequence of shape descriptions; the *reduced sequence* of S is obtained from S by replacing each maximal subsequence a^+ of S with the single element a . The *size* of S is the number of elements in its reduced sequence.

► **Lemma 7.** *Let S' be a sequence of shape descriptions from the feasible sets of every $G_{\nu'}$, with $\nu' \in N'$. We can decide whether S' is a shape sequence of G'_μ and compute the corresponding shape descriptions of G'_μ in $\mathcal{O}(r^3)$ time, where r is the size of S' . Furthermore there are $\mathcal{O}(r^2)$ computed shape descriptions of G'_μ .*

Let ν be a child of G_μ with $\nu \notin N'$, let s be a shape description of G_ν , and let G''_μ be the union of G_ν and G'_μ . We say that S' can be extended with s to a shape sequence S'' of G''_μ if S'' is a shape sequence of G''_μ , s belongs to S'' , and removing s from S'' we obtain S' .

► **Lemma 8.** *Let S' be a shape sequence of G'_μ . Given a shape description s of G_ν , we can decide whether S' can be extended with s to a shape sequence S'' of G''_μ and compute the corresponding shape descriptions of G''_μ in $\mathcal{O}(r^4)$ time, where r is the size of S' .*

Suppose that G_μ is upward planar and consider a uv -external upward planar embedding \mathcal{E}_μ of G_μ . We remove the special components of u and v and the normal components G_ν whose shape description labels the angle at u or v with -1 . There are at most two such components, as each one labels an internal angle at a pole with 1 . Let G'_μ be the subgraph of G_μ obtained after this removal; G'_μ is the *thin subgraph* of G_μ with respect to \mathcal{E}_μ . In the next lemma, if $w \in \{u, v\}$ is a top vertex then $\rho_w = \text{out}$, otherwise $\rho_w = \text{in}$.

► **Lemma 9.** *Let μ be a P -node such that G_μ is upward planar and let \mathcal{E}_μ be a uv -external upward planar embedding of G_μ such that the left-turn-number of G'_μ is c . Then the shape sequence of G'_μ with respect to \mathcal{E}_μ is $[s_1^+, s_2^*, s_3^*]$, with $s_1 = \langle c, -c, 1, 1, \rho_u, \rho_u, \rho_v, \rho_v \rangle$, $s_2 = \langle c - 2, -c + 2, 1, 1, \rho_u, \rho_u, \rho_v, \rho_v \rangle$, $s_3 = \langle c - 4, -c + 4, 1, 1, \rho_u, \rho_u, \rho_v, \rho_v \rangle$.*

Based on Lemma 9, our algorithm computes in three steps the shape descriptions of G_μ that match some fixed left-turn-number c_l and right-turn-number c_r . Let c'_l be equal to c_l or $c_l - 1$, depending on whether exactly one of u and v is a bottom vertex or not. For the first step, we consider all sequences $S' = [s_1^*, s_2^*, s_3^*]$ where $s_i = \langle c'_l - 2(i - 1), -c'_l + 2(i - 1), 1, 1, \rho_u, \rho_u, \rho_v, \rho_v \rangle$, for $i = 1, 2, 3$. For each of them we identify a maximal subgraph G'_μ of G_μ such that S' is a shape sequence of G'_μ . For each child ν_i of μ (with $i = 1, 2, \dots, k$), we check whether the feasible set \mathcal{F}_{ν_i} contains shape descriptions of S' in the order that they appear in S' ; if so, we choose it for G_{ν_i} . This greedy process does not necessarily produce the desired sequence S' . By reassigning at most two components of G'_μ either we get S' or no subgraph G'_μ has S' as its shape sequence.

For the second step, we focus on the children of μ that, when considering a shape sequence S' , have not been assigned a shape description so far. There are at most two such children, say ν and ν' , otherwise G_μ does not admit an upward planar embedding whose thin subgraph has S' as its shape sequence. Let s_ν (resp. $s_{\nu'}$) be a shape description in \mathcal{F}_ν (resp. $\mathcal{F}_{\nu'}$). Using Lemma 8 we compute all possible extensions of S' with s_ν and $s_{\nu'}$ to shape sequences of G_μ (in $\mathcal{O}(1)$ time since the size r of S' is at most 3). For every computed shape sequence S of G_μ we check whether it matches c_l and c_r . If so, we add to \mathcal{F}_μ all shape descriptions of G_μ that correspond to S (in $\mathcal{O}(1)$ time since the size r of S is at most 5). Otherwise, we proceed to the third step with S .

To complete the procedure, we perform a case analysis to handle situations where one or both of c_l and c_r are not matched. Intuitively, our goal is to find a component of the thin subgraph G'_μ , remove its current shape description from S and use another one from its feasible set at the beginning or at the end of the sequence in order to match c_l or c_r . If none of the components of G'_μ can be used for this purpose, we conclude that the pair c_l and c_r cannot be realized. Otherwise, using Lemma 7 (where the size r is at most 5), we compute all corresponding shape descriptions of G_μ and add them to \mathcal{F}_μ .

► **Lemma 10.** *Let μ be an P -node of T with k children. The feasible set \mathcal{F}_μ can be computed in $\mathcal{O}(\tau \cdot k)$ time.*

R-node. The R-nodes will be handled differently in Sections 6 and 7 depending on the parameter we use. To complete the description of our framework we introduce the notion of an R-node subprocedure. Formally, an *R-node subprocedure* is an algorithm which takes as

input an R-node μ of T and a mapping \mathcal{S}_μ which assigns each child of μ to its feasible set, and computes the feasible set \mathcal{F}_μ in at most $\alpha(G_\mu, \mathcal{S}_\mu)$ time. For a DAG G with SPQR-tree T , $\alpha(G) = \sum_{\text{R-node } \mu} \alpha(G_\mu, \mathcal{S}_\mu)$ is the *total time complexity* of the R-node subprocedure for G .

Root node. The root node r corresponds to an edge $e = (u, v)$ of G that lies on its outer face and has only one child μ with poles u and v . Since $G_\mu = G \setminus e$, node r can be treated as a P-node with poles u and v and two children; one of them is μ and the other one is a Q-node for the edge (u, v) . By Lemma 10, we can compute the feasible set of r in $\mathcal{O}(\tau)$ time.

► **Lemma 11.** *The feasible set \mathcal{F}_r of the root node r of T can be computed in $\mathcal{O}(\tau)$ time.*

Combining Lemmata 5, 6, 10 and 11, we obtain the following lemma.

► **Lemma 12.** *Let G be a biconnected DAG with n vertices and let T be an SPQR-tree of G rooted at a Q-node corresponding to an edge $e = (u, v)$. Let τ_{\min} and τ_{\max} be two given integer values, and let $\tau = \tau_{\max} - \tau_{\min} + 1$. Given an R-node subprocedure with total time complexity $\alpha(G)$, it is possible to compute in time $\mathcal{O}(\alpha(G) + \tau^2 \cdot n)$ the shape descriptions of every upward planar embedding with e on the outer face, such that the left- and right-turn-numbers of the pertinent graph of every node of T are in the range $[\tau_{\min}, \tau_{\max}]$.*

5 Extension to the Single-Connected Case

In this section, we establish the Interface Lemma, which reduces the task of solving UPWARD PLANARITY to the one of obtaining an R-node subprocedure, for *all* graphs including single-connected ones. To formalize the lemma, we call a digraph G $[\tau_{\min}, \tau_{\max}]$ -turn-bounded if every upward planar embedding of G has the following property: the pertinent graphs of any SPQR-tree of each biconnected component in G have left- and right-turn numbers in the range $[\tau_{\min}, \tau_{\max}]$.

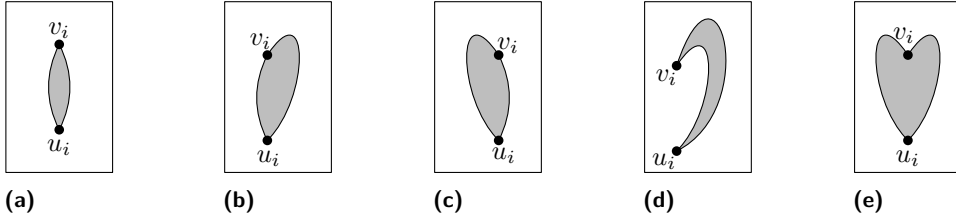
► **Lemma 13 (Interface Lemma).** *Let G be an n -vertex digraph, and τ_{\min}, τ_{\max} be integers such that G is $[\tau_{\min}, \tau_{\max}]$ -turn-bounded. Given an R-node subprocedure with total time complexity $\alpha(G)$, it is possible to determine whether G admits an upward planar embedding in time $\mathcal{O}(n(\alpha(G) + \tau^2 \cdot n))$ where $\tau = \tau_{\max} - \tau_{\min} + 1$.*

Note that, for a single-connected graph G , we define the total time complexity $\alpha(G)$ of an R-node subprocedure to be the sum of $\alpha(B)$ over all biconnected components B of G .

To give an intuition of the proof, consider a fixed rooting of the block-cut tree of G . The core of our algorithm is a procedure that, given suitable embeddings of leaf components containing the same cut-vertex, attaches these embeddings to an arbitrary upward planar embedding of the rest of the graph. This allows us to process the block-cut tree upwards: we iteratively verify that there exist desired embeddings for a group of leaf blocks via the biconnected algorithm (Lemma 12), and reduce to a smaller tree by removing these blocks.

6 An Algorithm Parameterized by the Number of Sources

Let G be an acyclic digraph with n vertices and σ sources, whose underlying graph is planar. In order to obtain an algorithm for UPWARD PLANARITY parameterized by σ , in view of Lemma 13, we devise an R-node subprocedure whose runtime depends on σ and, polynomially, on n . We hence assume that G is biconnected and that has been expanded. Let e^* be any edge of G ; we compute an SPQR-tree T of G rooted at the Q-node representing e^* in $\mathcal{O}(n)$ time [12, 22]. A key ingredient of our algorithm is the following.



■ **Figure 4** Shape descriptions of boring components.

► **Lemma 14.** *Let μ be a node of T , let u and v be the poles of μ , let σ_μ be the number of sources of G_μ different from its poles, and let \mathcal{E}_μ be any uv -external upward planar embedding of G_μ . The left- and right-turn-numbers of \mathcal{E}_μ are in the interval $[-2\sigma_\mu - 1, 2\sigma_\mu + 1]$. Furthermore, the size of the feasible set \mathcal{F}_μ of μ is at most $72\sigma_\mu + 54$.*

Let μ be an R-node of T with children ν_1, \dots, ν_k . Let u and v be the poles of μ , σ_μ be the number of sources of G_μ different from its poles; for $i = 1, \dots, k$, let u_i and v_i be the poles of ν_i and e_i be the virtual edge representing ν_i in the skeleton $\text{sk}(\mu)$ of μ . We give an algorithm that computes \mathcal{F}_μ from the feasible sets $\mathcal{F}_{\nu_1}, \dots, \mathcal{F}_{\nu_k}$ in $\mathcal{O}(\sigma 1.45^\sigma \cdot k \log^3 k)$ time.

We introduce two classifications of the components of G_μ . A component G_{ν_i} is *interesting* if it contains sources other than its poles, and *boring* otherwise. Because G has σ sources, at most σ components among $G_{\nu_1}, \dots, G_{\nu_k}$ are interesting, while any number of components can be boring. Second, a component G_{ν_i} is *extreme* if e_i is incident to a pole of μ and is incident to the face containing u and v of any planar embedding of $\text{sk}(\mu)$, and *non-extreme* otherwise. Note that there are four extreme components among $G_{\nu_1}, \dots, G_{\nu_k}$, because there are exactly two virtual edges incident to each of u and v in the considered face. We can order $G_{\nu_1}, \dots, G_{\nu_k}$ in $\mathcal{O}(k \log k)$ time so that all the extreme or interesting components come first.

Despite their name, boring components play an important role in our algorithm. A key feature is that a $u_i v_i$ -external upward planar embedding of a boring component G_{ν_i} can only have one of $\mathcal{O}(1)$ shape descriptions: the **sausage** $\langle 0, 0, 1, 1, \text{out}, \text{out}, \text{in}, \text{in} \rangle$, see Fig. 4a; the **inverted-sausage** $\langle 0, 0, 1, 1, \text{in}, \text{in}, \text{out}, \text{out} \rangle$, see Fig. 4a with u_i and v_i inverted; the **right-wing** $\langle 0, 1, 1, 0, \text{out}, \text{out}, \text{in}, \text{out} \rangle$, see Fig. 4b; the **inverted-right-wing** $\langle 1, 0, 0, 1, \text{out}, \text{in}, \text{out}, \text{out} \rangle$, see Fig. 4b with u_i and v_i inverted; the **left-wing** $\langle 1, 0, 1, 0, \text{out}, \text{out}, \text{out}, \text{in} \rangle$, see Fig. 4c; the **inverted-left-wing** $\langle 0, 1, 0, 1, \text{out}, \text{in}, \text{out}, \text{out} \rangle$, see Fig. 4c with u_i and v_i inverted; the **hat** $\langle -1, 1, 1, 1, \text{out}, \text{out}, \text{out}, \text{out} \rangle$, see Fig. 4d; the **inverted-hat** $\langle 1, -1, 1, 1, \text{out}, \text{out}, \text{out}, \text{out} \rangle$, see Fig. 4d with u_i and v_i inverted; the **heart** $\langle 1, 1, 1, -1, \text{out}, \text{out}, \text{out}, \text{out} \rangle$, see Fig. 4e; and the **inverted-heart** $\langle 1, 1, -1, 1, \text{out}, \text{out}, \text{out}, \text{out} \rangle$, see Fig. 4e with u_i and v_i inverted. Furthermore, we can prove that not all such shape descriptions can occur simultaneously in the feasible set of a node ν_i and that some shape descriptions are “better” than others. This allows us to assume that the feasible set of a node ν_i contains: only the sausage, or only the inverted-sausage, or only the left-wing and the right-wing, or only the inverted-left-wing and the inverted-right-wing, or only the hat and the inverted-hat, or only the heart, or only the inverted-heart, or only the heart and the inverted-heart.

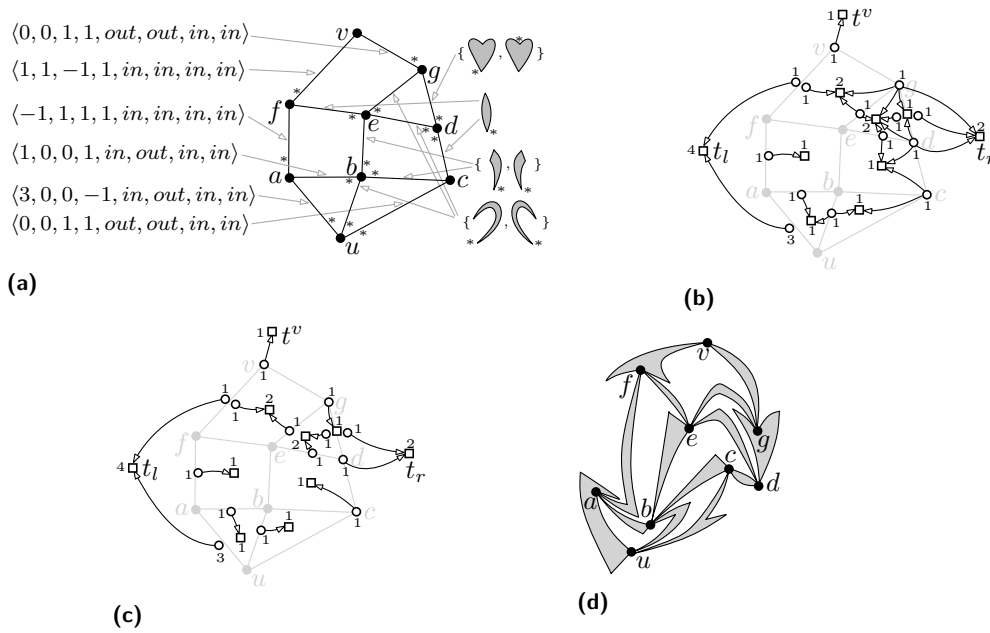
We test independently whether each shape description $s = \langle \tau_l, \tau_r, \lambda_u, \lambda_v, \rho_{l,u}, \rho_{r,u}, \rho_{l,v}, \rho_{r,v} \rangle$, where $\tau_l \in [-2\sigma_\mu - 1, 2\sigma_\mu + 1]$, $\tau_r \in [-\tau_l, -\tau_l + 4]$, $\lambda_u \in \{-1, 0, 1\}$, $\lambda_v \in \{-1, 0, 1\}$, $\rho_{l,u} \in \{\text{in}, \text{out}\}$, $\rho_{r,u} \in \{\text{in}, \text{out}\}$, $\rho_{l,v} \in \{\text{in}, \text{out}\}$, and $\rho_{r,v} \in \{\text{in}, \text{out}\}$ belongs to \mathcal{F}_μ or not. Note that $\tau_l \in [-2\sigma_\mu - 1, 2\sigma_\mu + 1]$ and $\tau_r \in [-\tau_l, -\tau_l + 4]$ can be assumed without loss of generality by Lemmata 14 and 3, respectively, thus the number of shape descriptions to be tested is in $\mathcal{O}(\sigma_\mu)$. We select shape descriptions $s_1 \in \mathcal{F}_{\nu_1}, \dots, s_h \in \mathcal{F}_{\nu_h}$ for the extreme or interesting components $G_{\nu_1}, \dots, G_{\nu_h}$ of G_μ . Clearly, the number ℓ of ways this selection can

be done is $\ell = \prod_{i=1}^h |\mathcal{F}_{\nu_i}|$; by exploiting the bound on $|\mathcal{F}_{\nu_i}|$ given by Lemma 14, we can prove that $\ell \in \mathcal{O}(1.45^\sigma)$. We also fix \mathcal{S}_μ to be a planar embedding of the skeleton $\text{sk}(\mu)$ of μ in which u and v are incident to the outer face. Since μ is an R-node, there are two such planar embeddings, which are flips of each other. The goal now becomes the one of testing whether G_μ admits a uv -external upward planar embedding \mathcal{E}_μ such that: (P1) the shape description of \mathcal{E}_μ is s ; (P2) for $i = 1, \dots, h$, the $u_i v_i$ -external upward planar embedding \mathcal{E}_{ν_i} of G_{ν_i} in \mathcal{E}_μ has shape description s_i ; and (P3) the planar embedding of $\text{sk}(\mu)$ induced by \mathcal{E}_μ is \mathcal{S}_μ . Then we have that s belongs to \mathcal{F}_μ if and only if this test is successful for at least one selection of the shape descriptions s_1, \dots, s_h and of the planar embedding \mathcal{S}_μ .

We now borrow ideas from an algorithm by Bertolazzi et al. [3] for testing the upward planarity of a digraph D with a prescribed planar embedding \mathcal{E} . The algorithm in [3] constructs a bipartite flow network $\mathcal{N}(S, T, A)$, where each source $s_w \in S$ corresponds to a switch vertex w of D , each sink $t_f \in T$ corresponds to a face f of \mathcal{E} , and A has an arc from s_w to t_f if w is incident to f . A unit of flow passing from s_w to t_f corresponds to a large angle at w in f . Each source supplies 1 unit of flow, each arc has capacity 1, and each sink t_f demands $n_f/2 - 1$ units of flow if f is an internal face of \mathcal{E} and $n_f/2 + 1$ if f is the outer face of \mathcal{E} , where n_f is the number of switch angles incident to f . Then D has an upward planar embedding which respects \mathcal{E} if and only if \mathcal{N} has a flow in which each sink is supplied with a number of units of flow equal to its demand.

After some preliminary checks, which ensure that the values $s, s_1, \dots, s_h, \mathcal{S}_\mu$ are “coherent” with each other, we also construct a flow network $\mathcal{N}(S, T, A)$. Note that the skeleton $\text{sk}(\mu)$ of our R-node μ has a prescribed planar embedding \mathcal{S}_μ . However, the edges of $\text{sk}(\mu)$ are not actual edges, but rather virtual edges that correspond to components of G_μ . These components introduce new sources, sinks, and arcs in \mathcal{N} , and contribute to the demands of their incident faces. As we have already fixed the shape description s_i of each extreme or interesting component G_{ν_i} , we know the excess of large angles with respect to small angles “on the sides” of G_{ν_i} , as these are the first two values of s_i . These values introduce sources (if they are positive) and contribute to the demands of the faces of \mathcal{S}_μ incident to e_i . Handling non-extreme boring components is more challenging. Each boring component has at most two shape descriptions in its feasible set, however the number of such components is not, in general, bounded by a function of σ only, hence we cannot try all possible combinations for their shape descriptions. Rather, we plug the freedom of choosing a shape description for each non-extreme boring component directly into the flow network. For example, a component G_{ν_i} such that \mathcal{F}_{ν_i} contains the hat and the inverted-hat is modeled by a source with two incident arcs to the faces of \mathcal{S}_μ incident to e_i , reflecting the fact that each of the two shape descriptions provides a large angle in a different face incident to e_i . As another example, a component G_{ν_i} such that \mathcal{F}_{ν_i} contains the left-wing and the right-wing also provides a large angle in a different face incident to e_i depending on the choice of the shape description, however in this case the choice might also affect whether a pole of the component creates a switch angle in a face of \mathcal{S}_μ or not, which affects the demand of the face. This is solved either by “ignoring” the component, or by transferring its effect to an adjacent non-switch vertex.

Figure 5 shows an example of the construction of \mathcal{N} . We have that \mathcal{N} has $\mathcal{O}(k)$ nodes and arcs. We test whether every sink has a non-negative demand and whether \mathcal{N} admits a flow in which every sink receives an amount of flow equal to its demand. The latter can be done in $\mathcal{O}(k \log^3 k)$ time by means of an algorithm by Borradaile et al. [4]. We conclude that G_μ admits a uv -external upward planar embedding satisfying Properties P1–P3 if and only if the tests are successful. This leads to the following.



■ **Figure 5** The construction of a flow network \mathcal{N} that allows us to determine whether a shape description $s = \langle 1, 0, -1, 0, in, out, in, in \rangle$ belongs to \mathcal{F}_μ . (a) shows the input: a shape description s_i for each extreme or interesting component G_{ν_i} of G_μ and the feasible set \mathcal{F}_{ν_i} for each non-extreme boring component G_{ν_i} of G_μ . (b) shows \mathcal{N} ; arc capacities are not shown (each of them is equal to the supply of the source of the arc). (c) shows a flow for \mathcal{N} in which every sink receives an amount of flow equal to its demand; each shown arc is traversed by a flow equal to its capacity. (d) shows a uv -external upward planar embedding of G_μ with shape description s corresponding to the flow.

► **Lemma 15.** *The feasible set \mathcal{F}_μ of an R-node μ of T can be computed in $\mathcal{O}(\sigma 1.45^\sigma \cdot k \log^3 k)$ time, where k is the number of children of μ in T and σ is the number of sources of G .*

Lemmata 13 and 15 imply the following main result.

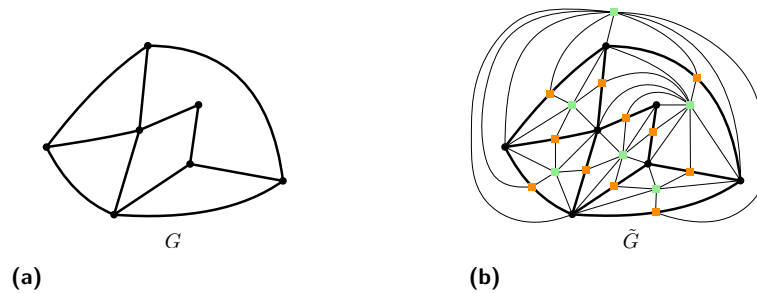
► **Theorem 16.** *UPWARD PLANARITY can be solved in $\mathcal{O}(\sigma 1.45^\sigma \cdot n^2 \log^3 n)$ time for a digraph with n vertices and σ sources.*

7 An Algorithm Parameterized by Treewidth

The aim of this section is to provide an R-node subprocedure which yields parameterized algorithms for UPWARD PLANARITY when parameterized by treewidth and treedepth. The idea behind this is to obtain a combinatorization of the task that is asked in the subprocedure. This will be done by extending the skeleton of the R-node with additional information, notably via a so-called *embedding graph*². The R-node subprocedure is then obtained by performing dynamic programming over the embedding graph. However, to obtain the desired runtime, we will first have to show that the embedding graph has bounded treewidth.

² Note that this notion differs from the embedding graphs used in recent drawing extension problems [16, 17]; unlike in those problems, here it seems impossible to use Courcelle’s Theorem [9].

A Combinatorial Representation of the Skeleton. Let G be a connected graph with a planar embedding \mathcal{G} , and let F be the set of faces of \mathcal{G} . Let G^- be the graph obtained from G by subdividing each edge e once, creating the vertex v_e . We define the *embedding graph* \tilde{G} of G as the graph obtained from G^- by adding a vertex f for each face in F , and connecting f to each vertex in G^- incident to f . Observe that \tilde{G} is tripartite, and we call the three sets of vertices that occur in the definition of $V(\tilde{G})$ the *true vertices*, *face-vertices* and *edge-vertices* of \tilde{G} , respectively. An illustration is shown in Fig. 6.



■ **Figure 6** (a) A planar graph G . (b) The embedding graph \tilde{G} of G . True-, face-, and edge-vertices are shown in black, green, and orange, respectively.

Our aim in this section is to show that $tw(\tilde{G})$ is linearly bounded by $tw(G)$. To do so, we identify the faces that are, in some sense, “relevant” for a bag in a tree decomposition of G^- , and prove that (1) the number of such relevant faces is linearly bounded by the width of that decomposition and (2) adding these faces to the decomposition of G^- results in a tree-decomposition of \tilde{G} . We can then prove:

► **Theorem 17.** *Let G be a graph with a planar embedding of treewidth k where $k \geq 1$. Then the embedding graph \tilde{G} has treewidth at most $11k - 4 \in \mathcal{O}(k)$.*

Problem Reformulation. Our second task is to formulate the problem we have to solve on a given embedding graph. First of all, the R-node subprocedure required by Lemma 13 can be straightforwardly reduced to the task of checking whether a specific shape description ψ can be achieved at the R-node. This reduction takes at most $\mathcal{O}(\tau)$ time by Lemma 4. At this point, the input consists of (1) an R-node μ of T with skeleton H , (2) a mapping \mathcal{S}_μ which assigns each virtual edge in H to its feasible set, (3) a bound κ on the treewidth of the embedding graph \tilde{H} obtained from H , and (4) a target shape description ψ .

The combinatorial reformulation we obtain can be stated as follows: Determine if there exists an *angle mapping* α and *shape selector* β which is *valid*, where

- an angle mapping α maps each switch vertex $v \in V(\tilde{H})$ to a vertex in $N_{\tilde{H}}(v)$; intuitively, this describes where the large angle at v is in the upward planar embedding of the pertinent graph (this may be in a face between two virtual edges—and α maps v to the corresponding face vertex – or in a virtual edge—and α maps v to that virtual edge),
- a shape selector β maps each edge-vertex v_e obtained from the virtual edge e of H to a shape description that occurs in a feasible set in the range of $\mathcal{S}_\mu(e)$, and
- intuitively, a pair (α, β) is valid if it satisfies three Validity Conditions: (1) all face-vertices receive the correct number of small and large angles from α and β , (2) for each true-vertex v and adjacent edge-vertex w , $\alpha(v)$ is consistent with the requirements of the shape selected by $\beta(w)$, and (3) the shape of the outer face is consistent with ψ .

► **Lemma 18.** *There is an upward planar embedding of G_μ with the shape description ψ if and only if there is a valid pair (α, β) .*

Finding Valid Pairs Using Treewidth. At this point, what is left to do is solve this combinatorial problem. For the runtime analysis of the algorithm we will develop, we let ζ be the maximum over $n_1(f)$ and $n_{-1}(f)$ (see Theorem 1), over all faces f of all possible planar embeddings of the pertinent graph G_μ of μ . Recalling that no path in G can have length greater than $2^{td(G_\mu)}$ [27], we obtain:

► **Observation 19.** $\zeta \leq V(G_\mu)$, and moreover $\zeta \leq 2^{td(G_\mu)}$.

We can now design a dynamic program that solves the task at hand. The program computes sets of records for each node of a tree-decomposition in a leaf-to-root fashion, where each record is a tuple of the form `(angle, shape, score, left, right)` where `angle` and `shape` contain snapshots of α and β in the given bag, respectively; `score` keeps track of the sum of large and small angles for each face in the given bag; and `left, right` store information about the left-and right-turn-numbers of the outer face.

► **Lemma 20.** *There is an algorithm that runs in time $\zeta^{\mathcal{O}(tw(H))} \cdot (|V(H)| + |\mathcal{S}_\mu|)$ and either computes a valid pair, or correctly determines that no such pair exists.*

We now have an R-node subprocedure that runs in XP-time parameterized by treewidth and fixed-parameter time parameterized by treedepth. By invoking Lemma 13, we conclude:

► **Theorem 21.** *It is possible to solve UPWARD PLANARITY in time $n^{\mathcal{O}(tw(G))}$ and time $2^{\mathcal{O}(td(G)^2)} \cdot n^2$, where n is the number of vertices of the input digraph G .*

8 Concluding Remarks

The presented results show that the combination of SPQR-trees with parameterized techniques is a promising algorithmic tool for geometric graph problems. Indeed, for the case of upward planarity, our framework allows us to reduce the general problem to a similar one on 3-connected graphs, at which point it is possible to use parameter-specific approaches such as dynamic programming or flow networks to obtain a solution. We believe not only that the framework developed here can help obtain other algorithms for UPWARD PLANARITY, but that the idea behind the framework can be adapted to solve other problems of interest as well – a candidate problem in this regard would be constrained level planarity testing [6].

All algorithms and arguments given within this paper are constructive and can be extended to output an upward planar drawing for each yes-instance of UPWARD PLANARITY. An open problem is whether UPWARD PLANARITY is W[1]-hard when parameterized by treewidth, or fixed-parameter tractable. Another question is whether the fixed-parameter tractability of UPWARD PLANARITY parameterized by the number of sources can be lifted to parameterizing by the maximum turn number of a face in the final drawing.

References

- 1 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Yota Otachi. Grundy distinguishes treewidth from pathwidth. In *28th Annual European Symposium on Algorithms, ESA 2020*, volume 173 of *LIPICs*, pages 14:1–14:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ESA.2020.14.
- 2 Paola Bertolazzi, Giuseppe Di Battista, Carlo Mannino, and Roberto Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM J. Comput.*, 27(1):132–169, 1998.

- 3 Paola Bertolazzi, Giuseppe Di Battista, Giuseppe Liotta, and Carlo Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(6):476–497, 1994.
- 4 Glencora Borradaile, Philip N. Klein, Shay Mozes, Yahav Nussbaum, and Christian Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. *SIAM J. Comput.*, 46(4):1280–1303, 2017.
- 5 Guido Brückner, Markus Himmel, and Ignaz Rutter. An SPQR-tree-like embedding representation for upward planarity. In Daniel Archambault and Csaba D. Tóth, editors, *27th International Symposium on Graph Drawing and Network Visualization, GD 2019*, volume 11904 of *Lecture Notes in Computer Science*, pages 517–531. Springer, 2019.
- 6 Guido Brückner and Ignaz Rutter. Partial and constrained level planarity. In Philip N. Klein, editor, *28th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pages 2000–2011. SIAM, 2017.
- 7 Hubert Y. Chan. A parameterized algorithm for upward planarity testing. In Susanne Albers and Tomasz Radzik, editors, *12th Annual European Symposium on Algorithms, ESA 2004*, volume 3221 of *Lecture Notes in Computer Science*, pages 157–168. Springer, 2004.
- 8 Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Robert Ganian, Chrysanthi N. Raftopoulou, and Kirill Simonov. Parameterized algorithms for upward planarity. *CoRR*, abs/2203.05364, 2022. URL: <https://arxiv.org/abs/2203.05364>.
- 9 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.
- 10 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 11 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.
- 12 Giuseppe Di Battista and Roberto Tamassia. On-line planarity testing. *SIAM J. Comput.*, 25(5):956–997, 1996.
- 13 Walter Didimo, Francesco Giordano, and Giuseppe Liotta. Upward spirality and upward planarity testing. *SIAM J. Discret. Math.*, 23(4):1842–1899, 2009.
- 14 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 15 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.
- 16 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending partial 1-planar drawings. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, volume 168 of *LIPICs*, pages 43:1–43:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- 17 Robert Ganian, Thekla Hamm, Fabian Klute, Irene Parada, and Birgit Vogtenhuber. Crossing-optimal extension of simple drawings. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021*, volume 198 of *LIPICs*, pages 72:1–72:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- 18 Robert Ganian, Petr Hliněný, Joachim Kneis, Daniel Meister, Jan Obdržálek, Peter Rossmanith, and Somnath Sikdar. Are there any good digraph width measures? *J. Comb. Theory, Ser. B*, 116:250–286, 2016.
- 19 Robert Ganian, Fabrizio Montecchiani, Martin Nöllenburg, and Meirav Zehavi. Parameterized Complexity in Graph Drawing (Dagstuhl Seminar 21293). *Dagstuhl Reports*, 11(6):82–123, 2021.
- 20 Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. In Roberto Tamassia and Ioannis G. Tollis, editors, *DIMACS International Workshop on Graph Drawing, GD '94*, volume 894 of *Lecture Notes in Computer Science*, pages 286–297. Springer, 1994.

- 21 Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001.
- 22 Carsten Gutwenger and Petra Mutzel. A linear time implementation of SPQR-trees. In Joe Marks, editor, *8th International Symposium on Graph Drawing, GD '00*, volume 1984 of *Lecture Notes in Computer Science*, pages 77–90. Springer, 2000.
- 23 Patrick Healy and Karol Lynch. Two fixed-parameter tractable algorithms for testing upward planarity. *Int. J. Found. Comput. Sci.*, 17(5):1095–1114, 2006. doi:10.1142/S0129054106004285.
- 24 John E. Hopcroft and Robert Endre Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973. doi:10.1137/0202012.
- 25 Michael D. Hutton and Anna Lubiw. Upward planar drawing of single source acyclic digraphs. In Alok Aggarwal, editor, *2nd Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, SODA 1991*, pages 203–211. ACM/SIAM, 1991.
- 26 Michael D. Hutton and Anna Lubiw. Upward planar drawing of single-source acyclic digraphs. *SIAM J. Comput.*, 25(2):291–311, 1996. doi:10.1137/S0097539792235906.
- 27 Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.
- 28 Achilleas Papakostas. Upward planarity testing of outerplanar dags. In Roberto Tamassia and Ioannis G. Tollis, editors, *DIMACS International Workshop on Graph Drawing, GD '94*, volume 894 of *Lecture Notes in Computer Science*, pages 298–306. Springer, 1994. doi:10.1007/3-540-58950-3_385.
- 29 Neil Robertson and Paul D. Seymour. Graph minors. III. Planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.
- 30 William T. Trotter and John I. Moore Jr. The dimension of planar posets. *J. Comb. Theory, Ser. B*, 22(1):54–67, 1977.

Finding Weakly Simple Closed Quasigeodesics on Polyhedral Spheres

Jean Chartier ✉

Univ. Paris Est Creteil, CNRS, LAMA, F-94010 Creteil, France

Arnaud de Mesmay ✉

LIGM, CNRS, Univ. Gustave Eiffel, ESIEE Paris, F-77454 Marne-la-Vallée, France

Abstract

A closed quasigeodesic on a convex polyhedron is a closed curve that is locally straight outside of the vertices, where it forms an angle at most π on both sides. While the existence of a simple closed quasigeodesic on a convex polyhedron has been proved by Pogorelov in 1949, finding a polynomial-time algorithm to compute such a simple closed quasigeodesic has been repeatedly posed as an open problem. Our first contribution is to propose an extended definition of quasigeodesics in the intrinsic setting of (not necessarily convex) polyhedral spheres, and to prove the existence of a weakly simple closed quasigeodesic in such a setting. Our proof does not proceed via an approximation by smooth surfaces, but relies on an adaptation of the disk flow of Hass and Scott to the context of polyhedral surfaces. Our second result is to leverage this existence theorem to provide a finite algorithm to compute a weakly simple closed quasigeodesic on a polyhedral sphere. On a convex polyhedron, our algorithm computes a simple closed quasigeodesic, solving an open problem of Demaine, Hersterberg and Ku.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Quasigeodesic, polyhedron, curve-shortening process, disk flow, weakly simple

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.27

Related Version *Full Version*: <https://arxiv.org/abs/2203.05853> [11]

Funding This research was partially supported by the ANR project Min-Max (ANR-19-CE40-0014), the ANR project SoS (ANR-17-CE40-0033) and the Bézout Labex, funded by ANR, reference ANR-10-LABX-58.

Acknowledgements We thank Francis Lazarus for insightful discussions, and Joseph O’Rourke and the anonymous reviewers for helpful comments.

1 Introduction

A geodesic is a curve on a surface, or more generally in a manifold, which is locally shortest. The study of geodesics on surfaces dates back at least to Poincaré [20] and led to a celebrated theorem of Lyusternik and Schnirelmann [18] proving that any Riemannian sphere admits at least three distinct simple (i.e., not self-intersecting) closed geodesics (while the initial proof of the theorem was criticized, the result is now well-established, see for example Grayson [15]). This bound is tight, as showcased by ellipsoids.

In this article, we investigate closed geodesics in a polyhedral setting. In such a setting, the following relaxed notion is key: a *quasigeodesic* is a curve such that the angle is at most π on both sides at each point of the curve. In 1949, Pogorelov [19] proved the existence of three simple (i.e., non self-intersecting) and closed quasigeodesics on any convex polyhedron. The proof is non-constructive and it was asked by Demaine and O’Rourke [13, Open Problem 24.24] whether one could compute such a closed quasigeodesic in polynomial time. Recent progress on this question was made by Demaine, Hersterberg and Ku [12] who provided the first algorithm to compute a closed quasigeodesic on a convex polyhedron, and their



© Jean Chartier and Arnaud de Mesmay;

licensed under Creative Commons License CC-BY 4.0

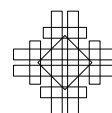
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 27; pp. 27:1–27:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



algorithm runs in pseudo-polynomial time. However, their algorithm is ill-adapted to find closed quasigeodesics which are simple – this has remained an open problem [12, Open Problem 1]. Furthermore, as they note, for this problem, “even a finite algorithm is not known or obvious”: indeed there is no known upper bound on the combinatorial complexity of a simple closed quasigeodesic (for example the number of times that it intersects each edge), so there is no natural brute-force algorithm. We refer to the extensive introduction of [12] for a panorama on the difficulties in finding closed quasigeodesics.

Our results. Our contributions in this article are two-fold.

First, we extend the theorem of Pogorelov to a non-convex and non-embedded setting. Precisely, we work in the abstract setting of compact *polyhedral spheres*, which consist of the following data: (1) a finite collection of Euclidean polygons, and (2) gluing rules between pairs of boundaries of equal length, so that the topological space resulting from the gluings is a topological sphere. A face, edge or vertex of a polyhedral sphere is respectively a polygon, an edge or a vertex of one of the polygons, and a vertex is *convex* (respectively *concave*) if the sum of the angles of the polygons around the vertex is at most 2π , respectively at least 2π . Let us emphasize that such a polyhedral sphere is not a priori embedded in \mathbb{R}^3 . In particular, edges of the triangles might not be shortest paths. This intrinsic description of non-smooth surfaces appears under various names in the literature, see, e.g., piecewise-linear surfaces [14] or intrinsic triangulations [21], and dates back to at least Alexandrov, who proved [2, Chapter 4] that when all the vertices are convex, such a polyhedral sphere is the metric structure of a unique convex polyhedron in \mathbb{R}^3 (see [17] for an algorithmic version of this result). In the non-convex case, a celebrated theorem of Burago and Zalgaller [8], shows that one can always find a piecewise-linear isometric embedding of a compact polyhedral sphere into \mathbb{R}^3 , but it might require a large number of subdivisions and the proof has to our knowledge not been made algorithmic.

Note that by definition, a polyhedral sphere is locally Euclidean at every point that is not a vertex. We propose the following generalization of the definition of quasigeodesics to a polyhedral sphere S : a closed quasigeodesic is a closed curve that is locally a straight line around any point that is not a vertex, and that is locally a pair of straight lines around a vertex, forming an angle *at most* π on each side if the vertex is convex, and forming an angle *at least* π on each side if the vertex is concave. A closed curve $\gamma : \mathbb{S}^1 \rightarrow S$ is *simple* if it is injective, and is *weakly simple* if it is a limit of simple curves (see Section 2 for details).

Our first theorem shows the existence of a weakly simple closed quasigeodesic of controlled length on a polyhedral sphere. We denote by M the *edge-sum* of S , which we define as the sum of the lengths of the edges of an iterated barycentric subdivision of a triangulation of S .

► **Theorem 1 (Existence).** *Let S be a polyhedral sphere and denote by M its edge-sum. There exists a weakly simple closed quasigeodesic of length at most M .*

The original proof of Pogorelov in the convex case proceeds by first approximating the polyhedron with smooth surfaces, and then taking the limit of the simple closed geodesics on the smooth surfaces, whose existence is guaranteed by the Lyusternik–Schnirelmann theorem. The proof technique for that latter theorem, originating from the work of Birkhoff [5], goes roughly as follows: we consider *sweep-outs*, i.e., a family of simple closed curves sweeping the polyhedron from one point to another point (see Section 2 for a precise definition), and consider the sweep-out where the longest curve has minimal length. Then, by applying a *curve-shortening process*, one can use this optimal sweep-out to find simple closed geodesics. This last step is notoriously perilous [3, 4, 15], hence the tumultuous history of the Lyusternik–Schnirelmann theorem. Our proof proceeds by working directly on the polyhedral sphere

and we prove the existence of a weakly simple closed quasigeodesic using a similar technique based on sweep-outs. Our key technical contribution is to rely on a curve-shortening process that is well-adapted to the polyhedral structure of the problem: we adapt the *disk flow* originally designed by Hass and Scott [16] for Riemannian surfaces so as to handle the disks formed by the stars of vertices in a seamless way. We are hopeful that this polyhedral variant of the disk flow could find further applications in the study of quasigeodesics.

Theorem 1 provides, in addition to the existence of a weakly simple closed quasigeodesic, a bound on its length. Our second result is to leverage this bound in order to control the combinatorics of the quasigeodesic, which allows us to design a finite algorithm to compute a weakly simple closed quasigeodesic on a polyhedral sphere.

► **Theorem 2 (Algorithm).** *Given a polyhedral sphere S , we can compute a weakly simple closed quasigeodesic in time exponential in n and $\lceil M/h \rceil$, where n is the number of vertices of S , M is its edge-sum, and h is the smallest altitude over all triangles of some triangulation of S .*

Note that a bound on the length of a quasigeodesic does not translate directly into a bound on the number of times that it crosses each edge of the polyhedral sphere, as these crossings could happen arbitrarily close to vertices, and thus contribute an arbitrarily small length. Our proof of Theorem 2 investigates the local geometry of quasigeodesics around vertices to show that this does not happen too much, and that one can indeed bound the multiplicity of each edge. Then, our algorithm guesses the correct combinatorics of the simple closed quasigeodesic and checks in polynomial time that it is realizable.

Our proof techniques for Theorem 1 only provide the existence of weakly simple quasigeodesics instead of simple quasigeodesics. We believe this to be a necessary evil in any generalization to the non-convex case, as shortest paths accumulate on concave vertices, making it impossible to define a curve-shortening process in the neighborhood of those which preserves simplicity. However, when all the vertices are convex, the result of Pogorelov does show the existence of a (actually three) simple closed quasigeodesics, where we include as a degenerate simple case a curve connecting twice two vertices of curvature at least π . Furthermore, his proof also provides an upper bound on the length of this simple quasigeodesic, as we explain at the end of Section 4. Since our algorithm behind Theorem 2 only relies on such an upper bound on the length and on the (weak) simplicity of the sought after curve, we can also use it to compute simple closed quasigeodesics in the convex case. This solves Open Problem 1 of [12], but note that we are still a long way off a polynomial-time algorithm.

Some of the proofs have been omitted and are available in the full version [11].

2 Preliminaries

In this article, a *polyhedral sphere* is a finite collection of Euclidean polygons, and gluing rules for boundaries of the same length, so that the space obtained by identifying the boundaries of the polygons via the gluing rules is homeomorphic to a sphere. Such a sphere is naturally endowed with a metric which is locally Euclidean at every point except at the vertices of the polygons, where it might display a *conical singularity*: if the total angle of the polygons glued around that vertex is larger than 2π (respectively at most 2π), we say that the vertex is *concave* (respectively *convex*), and its *curvature* is the angular defect compared to 2π (which is thus negative for concave vertices). Given a (not necessarily convex) polyhedron described via the coordinates of its vertices in \mathbb{R}^3 , one can easily compute the underlying polygons and thus the structure as a Euclidean sphere. The reverse direction of embedding a

polyhedral sphere in \mathbb{R}^3 is significantly more intricate (see [17] for the convex case and [8] for the general case), hence our choice of the intrinsic model.

Triangulating each polygon defining a polyhedral sphere yields a *triangulated polyhedral sphere*. Furthermore, by doing up to two barycentric subdivisions in each triangle if necessary, we can assume that there are no loops nor multiple edges in this triangulation. Note that this triangulation and these subdivisions do not change the metric of the sphere and do not impact quasigeodesicity (see next paragraph). Therefore, for convenience, in this article we will always assume that our polyhedral spheres are triangulated and that they contain neither loops nor multiple edges, and we will denote such a sphere by S from now on. A *shelling* of a triangulated sphere S is an order (T_1, \dots, T_ℓ) on the triangles that S consists of so that for all $i \in [1, \ell - 1]$, $\bigcup_{k=1}^{k=i} T_k$ is homeomorphic to a 2-disk D^2 . It is well-known that all the triangulated spheres are shellable, for example because, by Steinitz’s theorem [22, Chapter 4] they form the 2-skeleton of a polytope, and those are shellable [6]. Throughout this article, we use the following notations for a polyhedral sphere: its vertices are denoted by p_1, \dots, p_n , its edges by e_1, \dots, e_m (or sometimes e_{ij} to emphasize the vertices that it connects to) and its triangles by T_1, \dots, T_ℓ . The order induced by the numbering of the triangles is a shelling order. The *star* of vertex p_i , denoted by \mathcal{C}_i is the union of the triangles T_k having p_i for common vertex, identified along the edges adjacent to p_i . It is *convex* (resp. *concave*) if p_i is (but note that the shortest path in S between two points of a convex star is not necessarily contained in that star). We optionally rename the vertices of P to have $p_1 \in T_1$ and $p_n \in T_\ell$. Finally, we denote by M the sum of the lengths of the edges of S , and by h the smallest altitude of all the triangles in S . Note that h is a lower bound on the distance between any two vertices. For γ an edge or a curve on S , we denote by $L(\gamma)$ its length.

A *closed curve* c on S is a continuous map $c : \mathbb{S}^1 \rightarrow S$. A closed curve is *piecewise-linear* if it is locally straight except at a finite number of points.

► **Definition 3.** *A closed curve is a quasigeodesic if it is locally straight around every point of S that is not a vertex, and around a vertex it forms an angle at most (respectively at least) π on both sides if the vertex is convex (respectively concave).*

We emphasize that this definition is non-standard in the non-convex case, where it is sometimes simply forbidden for a quasigeodesic to go through a concave vertex [13]. Note that a quasigeodesic is straight around a vertex with zero curvature. A closed curve is *simple* if it is injective. Throughout this article, all the curves will always be parameterized at constant speed. We endow the space of piecewise-linear curves with the uniform convergence metric, i.e., $d(c_1, c_2) = \max_{t \in \mathbb{S}^1} d(c_1(t), c_2(t))$. A closed curve is *weakly simple* if it is a limit of simple curves for this metric: intuitively a weakly simple curve is a curve with tangencies but no self-crossings. We denote by \mathcal{P} the set of constant closed curves, i.e., closed curves c such that there exists $p \in S$ such that $\forall t \in \mathbb{S}^1, c(t) = p$.

We denote by Ω the space of rectifiable closed curves of length at most M . This space is compact for the uniform convergence metric, as can be shown via the Arzelà-Ascoli theorem, the bound on the length and the constant-speed parameterization providing equicontinuity (see for example [7, Theorem 2.5.14]). We denote by Ω^{pl} the subspace of Ω consisting of piecewise-linear and weakly simple closed curves. A *monotone sweep-out* of S is a continuous map $\beta : \mathbb{S}^2 \rightarrow S$, where \mathbb{S}^2 is seen as the quotient of the cylinder $[0, 1] \times \mathbb{S}^1$ by the relation which identifies the circles $(0, \mathbb{S}^1)$ and $(1, \mathbb{S}^1)$ to two points, and such that:

- $\beta(0, \cdot)$ and $\beta(1, \cdot)$ belong to \mathcal{P} , i.e., are two constant closed curves on S ,
- β has topological degree one,

- for $s \in (0, 1)$, each fiber $\beta(s, \cdot) : \mathbb{S}^1 \rightarrow S$ belongs to Ω^{pl} , and
- the sweep-out is monotone, i.e., if D_s denotes the disk to the left of $\beta(s, \cdot)$, the disks D_s are nested: $D_s \subseteq D_{s'}$ for $s' > s$.

The requirement on the topological degree informally means that each point is covered once by the sweep-out ; it is there to prevent trivial sweep-outs (for example constant at a point). It can be replaced by the requirement that the starting and endpoints are distinct. The monotonicity corresponds to the third condition, and typical sweep-outs in the literature do not assume it (see [9]), but in this paper we will only use monotone sweep-outs and thus for simplicity we will henceforth drop the word monotone. The *width* of a sweep-out is the length of the longest fiber. We denote by \mathcal{B} the space of sweep-outs.

The algorithm underlying Theorem 2 has complexity exponential in $\lceil M/h \rceil$, i.e., it depends on the actual values of the lengths of the boundaries of the polygons. Therefore, we do not work on a real RAM model and rely rather on a word RAM model, which is powerful enough to express all the operations that we require: see for example [12, Section 2] for a description of the $O(1)$ -Expression RAM model which can be encoded in the word RAM model and allows for a restricted notion of real numbers and algebraic operations thereon.

3 Disk flow and sweep-outs

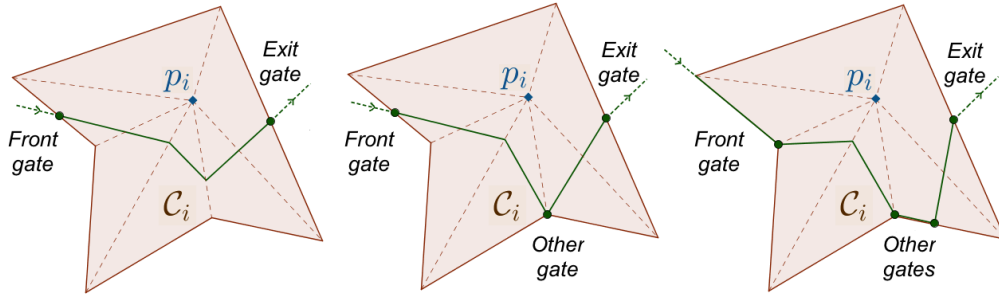
We start by describing a monotone sweep-out of controlled width.

► **Lemma 4.** *Let S be a triangulated polyhedral sphere of edge-sum M . There exists a monotone sweep-out of S of width at most M .*

This lemma is proved by sweeping triangles one by one, in the order prescribed by a shelling order of the sphere S .

The disk flow. We define here a curve-shortening process that we call the disk flow, which is an iterative process Φ shortening locally a curve in Ω^{pl} successively in each star \mathcal{C}_i , with the key property that the only fixed points of Φ are quasigeodesics or trivial curves. In a second step, we will extend Φ into a map $\hat{\Phi}$ that acts on monotone sweep-outs, which will require interpolating at the points where Φ is discontinuous. This disk flow is directly inspired by the work of Hass and Scott [16] who defined an analogous flow on Riemannian surfaces. The key difference with their setup is that the star \mathcal{C}_i around a convex vertex is not strongly convex (i.e. there is no uniqueness of shortest paths), which causes additional tears when extending Φ to sweep-outs and thus requires further operations. Furthermore, instead of working with very small convex disks as they are doing, we work directly with the stars \mathcal{C}_i as we strive to preserve curves whose piecewise-linear structure matches that of S . This requires us to deal with tangencies with the boundaries of stars in a different manner.

Let c be a curve in Ω^{pl} and let \mathcal{C}_i be a star crossed by $c \in \Omega^{pl}$. An *arc* of \mathcal{C}_i is a restriction of c whose image is a connected component of $\mathcal{C}_i \cap \text{Im}(c)$. Let γ be an arc of \mathcal{C}_i , from a closed curve c . The points $\gamma(t_0) = c(t_0) \in \partial\mathcal{C}_i$ such that $c([t_0 - \varepsilon, t_0))$ or $c((t_0, t_0 + \varepsilon])$ is contained in the interior of \mathcal{C}_i for a small enough $\varepsilon > 0$, are called the *gates* of γ . Note that two kinds of arcs have no gates: loops strictly inside the star and arcs never meeting the interior of the star. Unless γ is included in \mathcal{C}_i , the orientation of \mathbb{S}^1 naturally designates a first gate, denoted by **front**(γ), and a final gate, denoted by **exit**(γ). The gates can give access to the interior of the star for values of t greater (resp. less) than t_0 – we say that the gate is open to the right (resp. to the left). A gate can be open to the right and to the left. Thus, front gates are open to the right and exit gates are open to the left. Figure 1 illustrates different possible sequences of gates.



■ **Figure 1** Three examples of sequences of gates.

Relative to two gates A and B and independently of the path followed between A and B , we define the **right region** $\mathcal{C}_i^r(A, B)$ and the **left region** $\mathcal{C}_i^\ell(A, B)$ of the star, as being the two parts of \mathcal{C}_i whose union is \mathcal{C}_i and which intersect along the edges $[Ap_i]$ and $[p_iB]$. The orientation right/left is chosen compatible with that of c between the two gates. The angles of the regions at the p_i vertex are called the **right angle** $\theta_r(A, B)$ and the **left angle** $\theta_\ell(A, B)$.

► **Lemma 5.** Let c be a curve in Ω^{pl} . There exists a map $\Phi : \Omega^{pl} \rightarrow \Omega^{pl}$ satisfying the following two conditions :

- The only fixed points of Φ are quasigeodesics and constant curves.
- $L(\Phi(c)) \leq L(c)$, with equality if and only if c is a fixed point.

We stress that the map Φ is in general *not* continuous.

Proof. We define Φ as follows. Let c be a closed curve in Ω^{pl} . We pick an arbitrary order on the vertices of S , which induces an arbitrary order on the stars \mathcal{C}_i . The map Φ consists in repeating in this order a straightening process Φ_{loc}^i successively in each star. Consider in \mathcal{C}_i an arc γ of c . Note that between two of its consecutive gates, A open to the right and B open to the left, γ lies in \mathcal{C}_i .

If \mathcal{C}_i is convex, the straightening is defined as follows for each subset of γ between two consecutive gates (which by a slight abuse of notation we also denote by γ):

- If $p_i \in \gamma$ and if $\theta_r(A, B)$ and $\theta_\ell(A, B)$ are less than or equal to π , we replace γ by $[Ap_i] \cup [p_iB]$.
- If $p_i \notin \gamma$ and if $\theta_r(A, B)$ and $\theta_\ell(A, B)$ are less than or equal to π , we replace γ by the shortest path between A and B staying in the same region relative to A and B .
- If $\theta_r(A, B)$ (resp. $\theta_\ell(A, B)$) is strictly greater than π , we replace γ by the shortest path between A and B in \mathcal{C}_i^ℓ (resp. \mathcal{C}_i^r).

If \mathcal{C}_i is concave, the straightening is defined as follows:

- If $\theta_r(A, B)$ and $\theta_\ell(A, B)$ are at least π , and even if $p_i \notin \gamma$, we replace γ by $[Ap_i] \cup [p_iB]$.
- If $\theta_r(A, B)$ (resp. $\theta_\ell(A, B)$) is strictly less than π , we replace γ by the shortest path between A and B in \mathcal{C}_i^r (resp. \mathcal{C}_i^ℓ).

In case $\gamma = c$ is strictly included in the interior of \mathcal{C}_i , then $\Phi_{loc}^i(c) = 0$, where 0 denotes an arbitrary constant curve based at a point p in \mathcal{C}_i .

We denote by Φ_{loc}^i , relative to a given star \mathcal{C}_i , the straightening process described above, applied in this star to each arc of a closed curve $c \in \Omega^{pl}$. Then Φ is defined as the concatenation $\Phi := \circ_{i=1}^n \Phi_{loc}^i$. Let us first show that Φ has values in Ω^{pl} , note that it suffices to prove it for Φ_{loc}^i . It is immediate that the image under Φ_{loc}^i is piecewise-linear. In order

to prove that the image is weakly simple, we look at the case of two arcs of the same closed curve c in a star, one delimited by two gates A and B , the other delimited by two gates A' and B' . As c belongs to Ω^{pl} , the two arcs do not cross, so they delimit a band in the star. If Φ_{loc}^i sends both arcs to the same side of p_i , then their images form two shortest paths in the same region and do not intersect. If Φ_{loc}^i sends the two arcs on opposite sides of p_i , a configuration where the two arcs cross twice is impossible because the angles $\theta_r(A, B)$ and $\theta_r(A', B')$ on the one hand, and $\theta_\ell(A, B)$ and $\theta_\ell(A', B')$ on the other hand are arranged in the same order.

If c is a quasigeodesic, each of its arcs possibly behaves in two ways in the star it crosses: either it reaches and leaves the vertex in a straight line from and up to the boundary of the star, forming on each side an angle at most π in the convex case, or at least π in the concave case. Or it connects its gates via a shortest path, entirely contained in the more acute of the two regions that it induces. In both cases, the previous process does not change its trajectory. So Φ fixes the quasigeodesics. Conversely, if c is not a quasigeodesic, then either it does not take a shortest path through a face or in neighborhood of a transverse intersection with an edge, either it forms on the passage of a vertex an angle greater than π on one side. This will be straightened when applying Φ_{loc}^i in a star containing that face, edge, or vertex in its interior, and therefore c is not a fixed point of Φ_{loc}^i . By construction, since Φ_{loc}^i does not increase lengths, we have that $L(\Phi(c)) \leq L(c)$. Let us show that if $L(\Phi(c)) = L(c)$, then c is a quasigeodesic. If an arc of c is not fixed by Φ_{loc}^i in a star, while remaining on the same side of the vertex, then it loses length, because there is uniqueness of the shortest path within a (left or right) region of a star. On the other hand, if Φ_{loc}^i passes an arc on the other side of the vertex (or pushes it against the vertex), it is because its length exceeds $L([Ap_i]) + L([p_iB])$. So the arc loses at least this excess in length. Finally, since some Φ_{loc}^i decreases the length of a non-quasigeodesic c , such a c cannot be a fixed point of Φ . ◀

In this proof, we could have taken the simpler choice of always replacing an arc in a star by a shortest path, irrespective of the angle at the vertex. The more delicate choice that is made here is tailored so as to be able to extend Φ to sweep-outs in Lemma 8.

The following property will be useful.

► **Lemma 6.** *For all $\varepsilon > 0$, there exists $\eta > 0$ such that for any curve $c \in \Omega^{pl}$ and for any i , if $L(c) - L(\Phi_{loc}^i(c)) < \eta$, then $d_{\mathcal{H}}(c, \Phi_{loc}^i(c)) < \varepsilon$, where $d_{\mathcal{H}}$ denotes the Hausdorff distance.*

The following lemma shows that applying Φ iteratively to a curve either makes the curve trivial in finite time, or converges to a quasigeodesic. Note that the lemma is not as obvious as it might seem as Φ is not continuous on Ω^{pl} .

► **Lemma 7.** *Let $c \in \Omega^{pl}$. We consider the sequence of iterates of Φ , i.e., $(\Phi^j(c))_j$. If this sequence does not reach 0 in finite time, then it admits a subsequence converging to a quasigeodesic (with respect to the uniform convergence metric).*

We now explain how to apply the disk flow to a monotone sweep-out, so that it extends the action on each of the fibers.

► **Lemma 8.** *There exists a map $\hat{\Phi} : \mathcal{B} \rightarrow \mathcal{B}$ and a piecewise continuous injective map $\iota : [0, 1] \rightarrow [0, 1]$, such that*

$$\forall s \in [0, 1], \hat{\Phi}(\beta)(\iota(s), \cdot) = \Phi(\beta(s, \cdot)).$$

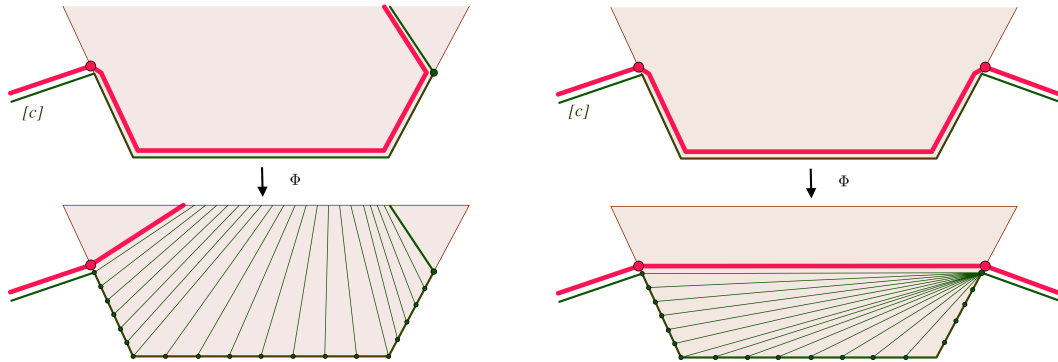
The map ι induces a surjection f that maps $[0, 1]$ on to $[0, 1]$, which continually extends ι^{-1} , with the property that $L(\hat{\Phi}(\beta)(s, \cdot)) \leq L(\beta(f(s), \cdot))$, with equality if and only if $\beta(f(s), \cdot)$ is a quasigeodesic.

Proof. Let β be a sweep-out in \mathcal{B} . We explain how to apply a local step $\hat{\Phi}_{loc}^i$ of the curve-shortening process to β . Then, as before, we will define $\hat{\Phi}$ as the concatenation $\circ_{i=1}^n \hat{\Phi}_{loc}^i$.

Before analyzing the effect of Φ_{loc}^i on β , we apply an artificial thickening of β which fills its “problematic” portions on the boundary of each star and is defined as follows. We call the *bare boundary* of \mathcal{C}_i the set of points of $\partial\mathcal{C}_i$ which are not the gates of any arc of a fiber of β crossing \mathcal{C}_i . Consider a connected component of the bare boundary of a certain star \mathcal{C}_i . It is fully contained in the image of at least¹ one fiber c of β that:

- either connects two gates which are neither a front gate nor an exit gate,
- or it connects a front or exit gate on one side only (see the green curve on Figure 2, top left),
- or it does not connect any gate (see the green curve on Figure 2, top right).

In all three cases, we can see that applying Φ_{loc}^i would induce a discontinuity around c . This is pictured in Figure 2, where one sees that the action of Φ_{loc}^i on the red curve and the green curve would be very different, despite them being arbitrarily close. We handle this discontinuity as follows. Case 1 will fit into the more general surgery described below, and thus is not addressed at this stage. In cases 2 and 3, the idea is to replace the parameter s of $c = \beta(s, \cdot)$ by a closed interval describing a collection of copies of c all identical (hence the artificial nature of this thickening), except that we drag artificially the position of the single extremal gate (case 2) or we add two new front/exit gates (case 3), one of which moves along $\partial\mathcal{C}_i$. In both cases, the new gates keep or gain an open character to the right or to the left. The aim of this operation is that the arcs of c between these new artificial gates will become straightened by Φ_{loc}^i , thus ensuring the continuity of Φ_{loc}^i at c (see Figure 2).



■ **Figure 2** We artificially add gates on bare edges to obtain interpolating curves in their neighborhoods.

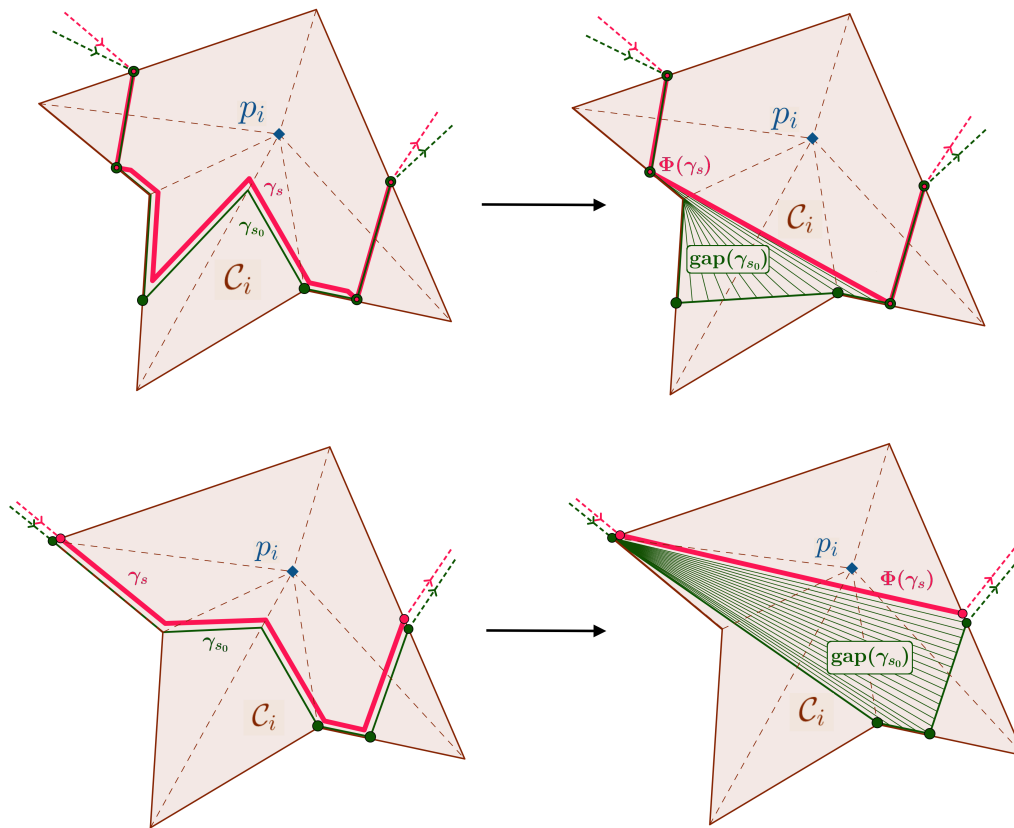
After this pre-processing, we consider the map $\beta' : [0, 1] \times \mathbb{S}^1 \rightarrow S$ defined by:

$$\forall s \in [0, 1], \beta'(s, \cdot) = \Phi_{loc}^i(\beta(s, \cdot)).$$

The discontinuity of Φ_{loc}^i on arcs within the star \mathcal{C}_i induces a finite number of tears in β' . These discontinuities occur around a convex vertex or when a fiber is tangent to the boundary of a star, and can be of the following four types.

¹ If there is an infinite number of them, they are parameterized in β by a closed interval. We then consider the representative closest to the interior of the star.

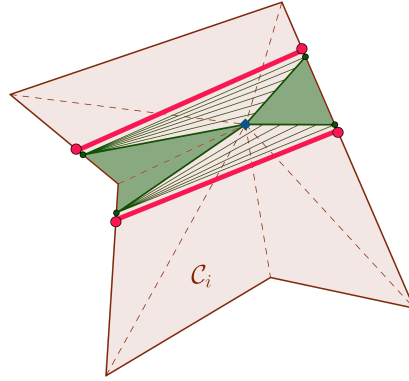
- Disappearance of one or more gates far from the vertex, see Figure 3.



■ **Figure 3** Disappearance of one or more gates far from the vertex: Two examples of interpolation. In the example in the bottom picture, the γ_{s_0} fiber has been added during the preprocessing and provided with an artificial gate. The missing part of the interpolation will be covered by the new gates induced in the preprocessing.

- Double tear around a convex vertex, see Figure 4.
- Single tear around a vertex, see Figure 5.
- Disappearance of interior curves, see Figures 6 and 7.

In all four cases, the discontinuities are filled by (1) blowing up the parameter space around a discontinuity point s to an interval $\text{gap}(s)$ and (2) adding interpolating curves in this interval $\text{gap}(s)$, one of which is $\Phi_{loc}^i(\beta(s, \cdot))$. We refer to the full version for a precise description of these interpolating procedures. Such interpolating curves are pictured in green in Figure 3, 4, 5, 6 and 7 and they are obtained by shortcutting $\beta(s, \cdot)$ using shortest paths, therefore all of them have length bounded by that of $\beta(s, \cdot)$. We define the map ι as the one sending s to the parameter corresponding to the fiber $\Phi_{loc}^i(\beta(s, \cdot))$, while the surjection f maps the entire interval $\text{gap}(s)$ to s (the maps ι and f are defined in the natural way outside of the discontinuities). Therefore we have defined a new map which we denote by $\hat{\Phi}_{loc}^i(\beta)$, whose parameter space is connected to that of β using the maps ι and f . As the $\hat{\Phi}_{loc}^i$ get composed to yield $\hat{\Phi}$, the maps ι and f are also composed in the natural way.



■ **Figure 4** Double tear around a convex vertex: Interpolation.

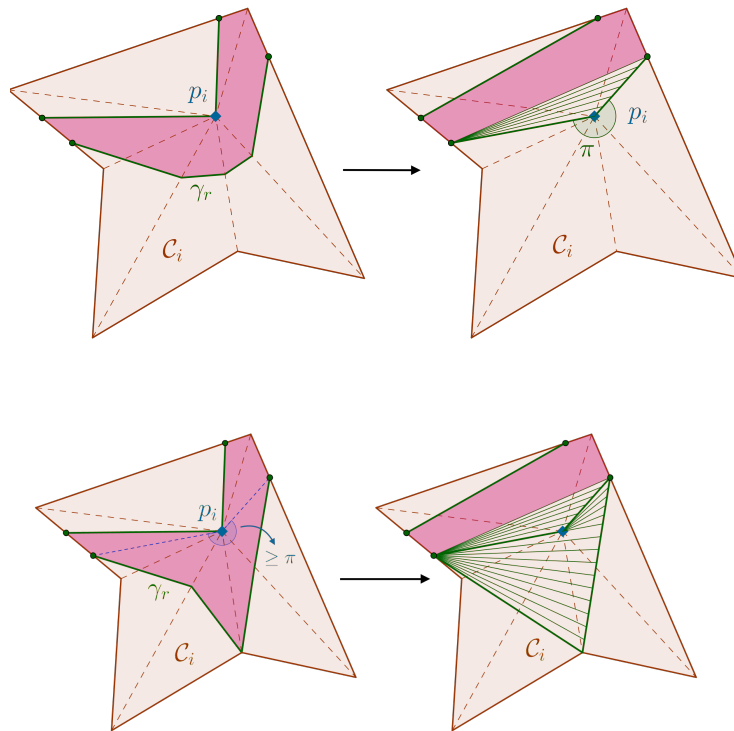
We argue that the resulting map is a monotone sweep-out. It starts and ends at trivial curves, and by construction each fiber is piecewise-linear. Furthermore, the disks defined by the fibers are nested, since the effect of $\hat{\Phi}_{loc}^i$ is restricted to the star \mathcal{C}_i , where the nesting of disks that was present in β is preserved, as the interpolated curves are put in between their interpolation targets. Generically, points are covered by the new sweep-out exactly once (since all the fibers can be slightly perturbed to be disjoint), thus the topological degree is one. Finally, since all the interpolating curves have length at most that of a curve it interpolates from, we have the inequality $L(\hat{\Phi}(\beta)(s, \cdot)) \leq L(\beta(f(s), \cdot))$, with equality if and only if $\beta(f(s), \cdot)$ is a quasigeodesic. ◀

► **Remark.** This proof showcases why our definition of quasigeodesic is the correct one for the disk flow to be appropriately defined on sweep-outs. If we had chosen more strict rules around convex vertices (for example only allowing curves with equal angles on both sides), we could have defined Φ in a more abrupt way by simply replacing arcs with shortest paths, thus ensuring that no arc through a vertex is fixed by the disk flow. However, this would have yielded tears around a convex vertex p in which our interpolating technique could not have worked, since no fiber of β' would be going through the vertex, and there would have been no way to add interpolating fibers of controlled length. In this sense, allowing for an angle at most π on both sides is the minimum angular spread allowing for the interpolation steps in the proof of Lemma 8 to work. For concave vertices, shortest paths between points on the boundary of a star \mathcal{C}_i might require the whole spread of angles at least π on both sides, hence this choice of definition.

4 Existence of a simple closed quasigeodesic

We are now ready to prove Theorem 1. At this stage, our proof follows the same lines as that of Hass and Scott [16, Theorem 3.11].

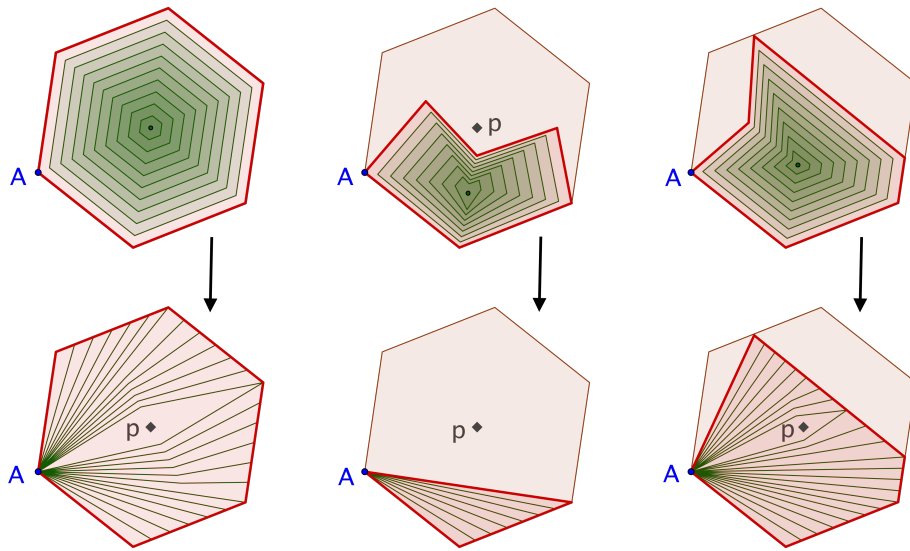
Proof of Theorem 1. Let β be the monotone sweep-out of \mathcal{B} of width at most M described by Lemma 4. We consider the sequence of sweep-outs $(\hat{\Phi}^j(\beta))_j$. For any $j \in \mathbb{N}$, the parameter space of $\hat{\Phi}^j(\beta)$ is the product of an interval $[0, 1]$ by \mathbb{S}^1 , the first factor being related to that of $\hat{\Phi}^{j-1}(\beta)$ via the surjection f_j of Lemma 8. Therefore, in order to track the history of a fiber in $\hat{\Phi}^j(\beta)$ under the action of $\hat{\Phi}$, we introduce the sequence of parameters $\mathcal{O}_j = (s_0, \dots, s_j)$ such that for all k between 0 and $j-1$: $s_k = f_k(s_{k+1})$. Each space of parameters describing



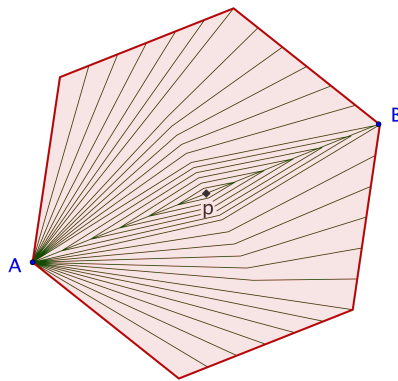
■ **Figure 5** Single tear around a vertex: Two examples of interpolation.

\mathcal{O}_j is homeomorphic to the interval $[0, 1]$ (via the trivial homeomorphism $(s_0, \dots, s_j) \mapsto s_j$), and we consider the projective limit \mathcal{I} of these intervals, which is thus also homeomorphic to an interval $[0, 1]$. An element of this projective limit therefore consists of an infinite sequence $\mathcal{O} = (s_0, s_1 \dots)$ such that for all k , $s_k = f_k(s_{k+1})$.

Let $\mathcal{O} = (s_1, s_2 \dots)$ be an element of \mathcal{I} , which thus corresponds to a family of curves $c_j := \hat{\Phi}^j(\beta)(s_j)$, and let us assume that all these curves are trivial for j bigger than some k . Then there is an open neighborhood of \mathcal{O} for which this is also the case, as a curve becomes trivial under the action of some $\hat{\Phi}_{loc}^i$ if and only if it is fully contained in the interior of a star. Therefore, the set $\mathcal{V}^k \subset \mathcal{I}$ of sequences of curves for which the k th curve is not trivial is a closed subset of \mathcal{I} . Furthermore, it is not empty, as otherwise some intermediate sweep-out after $\hat{\Phi}^{k-1}(\beta)$ would consist of only curves contained in the interior of some star and thus would miss some point of the sphere S , in contradiction with the requirement that a sweep-out be of topological degree one. Finally, we have the natural inclusion $\mathcal{V}^{k+1} \subset \mathcal{V}^k$ since if $\hat{\Phi}^{k+1}(\beta)(s_{k+1})$ is not trivial, then this is also the case for $\hat{\Phi}^k(\beta)(s_k)$. We can thus consider the intersection $\bigcap_{k \in \mathbb{N}} \mathcal{V}^k$ which is an infinite intersection of nested closed non-empty subsets of \mathcal{I} and is thus non-empty. An element in this intersection is a sequence $\mathcal{O}_\infty = (s_1, s_2 \dots)$ such that none of the curves $c_n = \hat{\Phi}^n(\beta)(s_n)$ is trivial. As Ω^{pl} is compact, we can extract from this sequence of curves a convergent subsequence c_k , which converges to a curve c_∞ . We claim that c_∞ is a weakly simple closed quasigeodesic of length at most M . The curve c_∞ is weakly simple because it is a limit of weakly simple curves. The bound on the length follows from the fact that by Lemma 4, the width of each of the sweep-outs $\hat{\Phi}^n(\beta)$ is at most M , and thus in particular c_∞ is a limit of curves of length at most M and thus has length at most M , since the length is a lower semi-continuous function on Ω .



■ **Figure 6** Disappearance of interior curves: Three examples of interpolation.



■ **Figure 7** Interpolating to fill the last hole when an interior curve disappears.

Finally, in order to prove that c_∞ is a quasigeodesic, the argument is identical to the one in the proof of Lemma 7, to which we refer. If c_∞ is not a quasigeodesic, there is one point p in its image which is not locally quasigeodesic, i.e., there are two points p_1 and p_2 in a small neighborhood in c_∞ such that p_1 , p and p_2 are not aligned, and if p is a vertex, the angle at p is disallowed by the curvature there. For k big enough, c_k will also have this property. Now, we consider a star \mathcal{C}_i which contains p in its interior. Here observe that Lemma 6 is also valid under the action of $\hat{\Phi}$, i.e., for interpolating curves: indeed, those are displaced even less than under the action of Φ . Therefore, c_k will have moved very little when Φ_{loc}^i acts on it, and thus this action will diminish its length by a fixed quantity that can be lower bounded based on c_∞ , which is impossible since the lengths of the c_k converge. ◀

Our techniques only guarantee the existence of a weakly simple closed quasigeodesic of length at most M . In contrast, in the convex case, Pogorelov [19] proved the existence of a simple closed quasigeodesic (where the degenerate case of two vertices of curvature at least π connected twice by a curve is allowed). The proof of Pogorelov works by approximating

a convex polyhedron by smooth surfaces, appealing to the Lyusternik-Schnirrelmann on the smooth surfaces to find simple closed geodesics, taking the limit of such simple closed geodesics and arguing that (1) the limit is a quasigeodesic and (2) it is simple. We argue that the same technique proves the existence of a simple closed quasigeodesic of length at most $M + \varepsilon$, for an arbitrarily small $\varepsilon > 0$. Indeed, the sweep-out that we describe on S naturally induces sweep-outs of width at most $M + \varepsilon$ on the approximating smooth surfaces that are close enough, and thus the first simple closed geodesic output by the Lyusternik-Schnirrelmann theorem in each of these surfaces has length at most $M + \varepsilon$. Taking the limit of those yields a simple closed quasigeodesic of length at most $M + \varepsilon$. We will use this result at the end of the next section.

5 An algorithm to compute a weakly simple closed quasigeodesic

In this section, we leverage the existence of a weakly simple closed quasigeodesic of length at most M proved in Theorem 1 in order to design an algorithm to find it.

Let S be a polyhedral sphere and denote by $\mathcal{E} = \{p_1, \dots, p_n, e_1, \dots, e_m\}$ be the set of vertices and open edges of S . To a closed curve $c : \mathbb{S}^1 \rightarrow S$, we associate the cyclic word $\mathcal{E}(c)$ whose successive letters are the elements of \mathcal{E} met by $c(t)$ as t moves around \mathbb{S}^1 (note that an edge can be either crossed or followed). Given a bound on the length of c , we want to derive a bound on the combinatorics of c , i.e., a bound on the length of $\mathcal{E}(c)$. This is hopeless without any assumption, as a curve spiraling around a vertex for an arbitrarily long time showcases. But when c is a weakly simple quasigeodesic, we can obtain such a bound. Indeed, our first observation is that a weakly simple quasigeodesic never spirals around a vertex.

► **Lemma 9.** *Let γ be a weakly simple closed quasigeodesic and \mathcal{C}_i be the open star of a vertex v_i of degree d_i . Then for any connected component α of $\gamma \cap \mathcal{C}_i$, the number of intersections of α with edges and vertices of \mathcal{C}_i is at most d_i .*

Proof. If α passes through the vertex v_i , then it exits on both sides tracing a straight line in one of the triangles of \mathcal{C}_i . This straight-line reaches directly the opposite edge of the triangle, therefore in this case the number of intersections of α with edges and vertices of \mathcal{C}_i is at most two.

If α does not pass through the vertex v_i , then let us denote by e the first edge adjacent to v_i that it crosses. Note that within a triangle of \mathcal{C}_i , by quasigeodesicity, α enters from one edge and does not backtrack, i.e., it escapes from another edge. Therefore, either α escapes from \mathcal{C}_i before crossing e again, in this case it crosses at most d_i edges, or it crosses e again. In the latter case, up to reversing orientation of α we can assume that the second crossing point is closer to v_i than the first crossing point. Tracing α after the second crossing point, we see that in each triangle that it enters, it cannot escape \mathcal{C}_i since, by weak simplicity, it cannot cross the previous edge that it traced, and is thus forced to continue spiraling around v_i indefinitely. This contradicts the assumption that γ is closed, finishing the proof. ◀

The following geometric lemma will come handy to bound the combinatorics of a closed simple quasigeodesic.

► **Lemma 10.** *Let Q be a Euclidean quadrilateral consisting of two Euclidean triangles glued along an edge. Then the distance between two opposite sides of Q is lower bounded by the smallest altitude of the two triangles.*

Combining Lemmas 9 and 10 yields the following proposition.

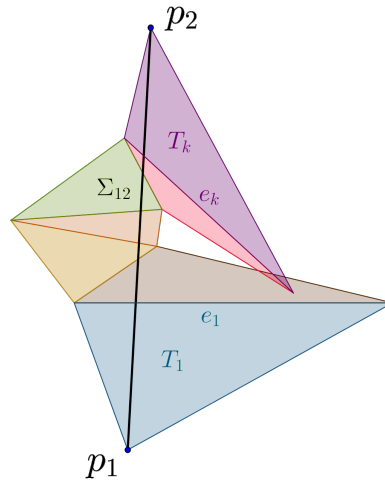
27:14 Finding Weakly Simple Closed Quasigeodesics on Polyhedral Spheres

► **Proposition 11.** *Let S be a polyhedral sphere, let M denote the sum of the edge-lengths of the triangles of S , let h denote the smallest altitude of the triangles of S , and let d be the maximum degree of a vertex in S . Then there exists a weakly simple closed quasigeodesic on S such that the length of $\mathcal{E}(\gamma)$ is bounded by:*

$$\eta_\gamma = \lceil (d + 1)M/h \rceil$$

We now have all the tools to prove Theorem 2.

Proof of Theorem 2. Let γ be a weakly simple closed quasigeodesic whose combinatorial complexity is controlled by η_γ as specified by Proposition 11. First, we observe that we can assume that this quasigeodesic meets a vertex (see full version).



■ **Figure 8** Unfolding a (tentative) quasigeodesic along the set of edges that it crosses.

Then, we guess the cyclic word w of size at most η_γ describing the combinatorics of γ , a weakly simple closed quasigeodesic going through at least one vertex. For each subword $p_1 e_1 \dots e_k p_2$ between two consecutive vertices, if e_1 is adjacent to p_1 , we simply check that the next letter is the other endpoint of p_1 . Otherwise, we first check that successive letters of that word are adjacent to a common triangle. Then we compute a local unfolding of the polyhedral sphere along the edges $e_1, e_2 \dots e_k$, i.e., we first place the triangle T_1 spanned by p_1 and e_1 , to which we attach along e_1 the triangle T_2 spanned by e_2 and e_3 , and so on until we reach the last triangle T_k spanned by e_k and p_2 . Now, in this unfolded picture, we trace the straight line Σ_{12} between p_1 and p_2 . There remains to check that the combinatorics of this straight line match those of the guessed word: in the first and last triangles, we check that Σ_{12} exits via or follows e_1 (or e_k), and in each other triangle T_i it suffices to check that the three vertices of T_i are on the sides of Σ_{12} prescribed by the edges e_i and e_{i+1} (i.e., if $e_i = ab$ and $e_{i+1} = bc$, then a and c should be one side of Σ_{12} while b should be on the other side). Then, we check that the angle between each pair $\Sigma_{i,i+1}, \Sigma_{i+1,i+2}$ is within the rules specified by the curvature at the vertex p_{i+1} . Finally, we check that this curve is weakly simple, for example via known algorithms [1, 10] or by brute-forcing in exponential time the choice of on which side two overlapping segments can be desingularized. If all the checks are positive, we have found the unique closed quasigeodesic matching the combinatorics of the word w , which is thus weakly simple. ◀

Finally, let us discuss how to find a simple closed quasigeodesic of bounded length in the case of a convex polyhedron. Following the discussion at the end of Section 4, Pogorelov’s theorem implies that there exists a simple closed quasigeodesic of length at most $M + \varepsilon$, for an arbitrarily small ε , and allowing as a “simple” closed quasigeodesic the degenerate case of a curve connecting twice two vertices of curvature at least π . This degenerate case is a weakly simple curve that will be found by our algorithm. For the non-degenerate case, the arguments of Proposition 11 apply verbatim to provide a bound on the combinatorics of some simple closed quasigeodesic γ . If this quasigeodesic goes through at least one vertex, the algorithm described just above finds it, and it is immediate to check that it is simple. If not, we can push it as in the proof of Theorem 2 to a weakly simple closed geodesic that goes through a vertex, and it will stay simple until it hits that vertex, where it will form an angle exactly π in the direction where it came from. Since the total angle at each vertex is at most 2π , this implies that this curve is either degenerate or simple, and in both cases it will be found by our algorithm.

References

- 1 Hugo A Akitaya, Greg Aloupis, Jeff Erickson, and Csaba D Tóth. Recognizing weakly simple polygons. *Discrete & Computational Geometry*, 58(4):785–821, 2017. doi:10.1007/s00454-017-9918-3.
- 2 Alexandr D Alexandrov. *Convex polyhedra*. Springer Science & Business Media, 2005.
- 3 Werner Ballmann. Der Satz von Lusternik und Schnirelmann. *Bonner Math. Schriften*, 102:1–25, 1978.
- 4 Werner Ballmann, Gudlaugur Thorbergsson, and Wolfgang Ziller. On the existence of short closed geodesics and their stability properties. In *Seminar On Minimal Submanifolds.(AM-103), Volume 103*, pages 53–64. Princeton University Press, 1983.
- 5 George David Birkhoff. *Dynamical systems*, volume 9 of *Colloquium Publications*. American Mathematical Soc., 1927. doi:10.1016/B978-044450871-3/50149-2.
- 6 Heinz Bruggesser and Peter Mani. Shellable decompositions of cells and spheres. *Mathematica Scandinavica*, 29(2):197–205, 1971. doi:10.7146/math.scand.a-11045.
- 7 Dmitri Burago, Yuri Burago, and Sergei Ivanov. *A course in metric geometry*, volume 33. American Mathematical Society, 2001.
- 8 Yuriy Dmitrievich Burago and Viktor Abramovich Zalgaller. Isometric piecewise-linear embeddings of two-dimensional manifolds with a polyhedral metric into \mathbb{R}^3 . *Algebra i analiz*, 7(3):76–95, 1995.
- 9 Erin Wolf Chambers, Gregory R Chambers, Arnaud de Mesmay, Tim Ophelders, and Regina Rotman. Constructing monotone homotopies and sweepouts. *Journal of Differential Geometry*, 119(3):383–401, 2021. doi:10.4310/jdg/1635368350.
- 10 Hsien-Chih Chang, Jeff Erickson, and Chao Xu. Detecting weakly simple polygons. In *Proceedings of the twenty-sixth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1655–1670. SIAM, 2014. doi:10.1137/1.9781611973730.110.
- 11 Jean Chartier and Arnaud de Mesmay. Finding weakly simple closed quasigeodesics on polyhedral spheres, 2022. arXiv:2203.05853.
- 12 Erik D Demaine, Adam C Hesterberg, and Jason S Ku. Finding closed quasigeodesics on convex polyhedra. In *36th International Symposium on Computational Geometry (SoCG 2020)*, pages 33:1–33:13. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. doi:10.1007/b137434.
- 13 Erik D Demaine and Joseph O’Rourke. *Geometric folding algorithms: linkages, origami, polyhedra*. Cambridge University Press, 2007. doi:10.1017/CB09780511735172.
- 14 Jeff Erickson and Amir Nayyeri. Tracing compressed curves in triangulated surfaces. *Discrete & Computational Geometry*, 49(4):823–863, 2013. doi:10.1007/s00454-013-9515-z.

27:16 Finding Weakly Simple Closed Quasigeodesics on Polyhedral Spheres

- 15 Matthew A Grayson. Shortening embedded curves. *Annals of Mathematics*, 129(1):71–111, 1989. doi:10.2307/1971486.
- 16 Joel Hass and Peter Scott. Shortening curves on surfaces. *Topology*, 33(1):25–43, 1994. doi:10.1016/0040-9383(94)90033-7.
- 17 Daniel Kane, Gregory N Price, and Erik D Demaine. A pseudopolynomial algorithm for Alexandrov’s theorem. In *Workshop on Algorithms and Data Structures*, pages 435–446. Springer, 2009. doi:10.1007/978-3-642-03367-4_38.
- 18 L Lyusternik and Lev Schnirelmann. Sur le problème de trois géodésiques fermées sur les surfaces de genre 0. *CR Acad. Sci. Paris*, 189(269):271, 1929.
- 19 Aleksei Vasil’evich Pogorelov. Quasi-geodesic lines on a convex surface. *Matematicheskii Sbornik*, 67(2):275–306, 1949. English translation in *American Mathematical Society Translations* 74, 1952.
- 20 Henri Poincaré. Sur les lignes géodésiques des surfaces convexes. *Transactions of the American Mathematical Society*, 6(3):237–274, 1905.
- 21 Nicholas Sharp, Yousuf Soliman, and Keenan Crane. Navigating intrinsic triangulations. *ACM Transactions on Graphics (TOG)*, 38(4):1–16, 2019. doi:10.1145/3306346.3322979.
- 22 Günter M Ziegler. *Lectures on polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer Science & Business Media, 2012. doi:10.1007/978-1-4613-8431-1.

Tight Lower Bounds for Approximate & Exact k -Center in \mathbb{R}^d

Rajesh Chitnis ✉

School of Computer Science, University of Birmingham, UK

Nitin Saurabh ✉

Indian Institute of Technology Hyderabad, Sangareddy, India

Abstract

In the discrete k -CENTER problem, we are given a metric space (P, dist) where $|P| = n$ and the goal is to select a set $C \subseteq P$ of k centers which minimizes the maximum distance of a point in P from its nearest center. For any $\epsilon > 0$, Agarwal and Procopiuc [SODA '98, Algorithmica '02] designed an $(1 + \epsilon)$ -approximation algorithm¹ for this problem in d -dimensional Euclidean space² which runs in $O(dn \log k) + \left(\frac{k}{\epsilon}\right)^{O(k^{1-1/d})} \cdot n^{O(1)}$ time. In this paper we show that their algorithm is essentially optimal: if for some $d \geq 2$ and some computable function f , there is an $f(k) \cdot \left(\frac{1}{\epsilon}\right)^{o(k^{1-1/d})} \cdot n^{o(k^{1-1/d})}$ time algorithm for $(1 + \epsilon)$ -approximating the discrete k -CENTER on n points in d -dimensional Euclidean space then the Exponential Time Hypothesis (ETH) fails.

We obtain our lower bound by designing a gap reduction from a d -dimensional constraint satisfaction problem (CSP) to discrete d -dimensional k -CENTER. This reduction has the property that there is a fixed value ϵ (depending on the CSP) such that the optimal radius of k -CENTER instances corresponding to satisfiable and unsatisfiable instances of the CSP is < 1 and $\geq (1 + \epsilon)$ respectively. Our claimed lower bound on the running time for approximating discrete k -CENTER in d -dimensions then follows from the lower bound due to Marx and Sidiropoulos [SoCG '14] for checking the satisfiability of the aforementioned d -dimensional CSP.

As a byproduct of our reduction, we also obtain that the exact algorithm of Agarwal and Procopiuc [SODA '98, Algorithmica '02] which runs in $n^{O(d \cdot k^{1-1/d})}$ time for discrete k -CENTER on n points in d -dimensional Euclidean space is asymptotically optimal. Formally, we show that if for some $d \geq 2$ and some computable function f , there is an $f(k) \cdot n^{o(k^{1-1/d})}$ time exact algorithm for the discrete k -CENTER problem on n points in d -dimensional Euclidean space then the Exponential Time Hypothesis (ETH) fails. Previously, such a lower bound was only known for $d = 2$ and was implicit in the work of Marx [IWPEC '06].

2012 ACM Subject Classification Theory of computation \rightarrow Design and analysis of algorithms

Keywords and phrases k -center, Euclidean space, Exponential Time Hypothesis (ETH), lower bound

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.28

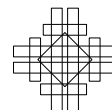
Related Version *Full Version*: <https://arxiv.org/abs/2203.08328> [7]

1 Introduction

The k -CENTER problem is a classical problem in theoretical computer science and was first formulated by Hakimi [22] in 1964. In this problem, given a metric space (P, dist) and an integer $k \leq |P|$ the goal is to select a set C of k centers which minimizes the maximum

¹ The algorithm of Agarwal and Procopiuc [2] also works for the non-discrete, i.e., continuous, version of the problem when C need not be a subset of P , but our lower bounds only hold for the discrete version.

² The algorithm of Agarwal and Procopiuc [2] also works for other metrics such as ℓ_∞ or ℓ_q metric for $q \geq 1$. Our construction also works for ℓ_∞ (in fact, some of the bounds are simpler to derive!) but we present only the proof for ℓ_2 to keep the presentation simple.



distance of a point in P from its nearest center, i.e., select a set C which minimizes the quantity $\max_{p \in P} \min_{c \in C} \text{dist}(p, c)$. A geometric way to view the k -CENTER problem is to find the minimum radius r such that k closed balls of radius r located at each of the points in C cover all the points in P . In most applications, we require that $C \subseteq P$ and this is known as the discrete version of the problem.

As an example, one can consider the set P to be important locations in a city and solving the k -CENTER problem (where k is upper bounded by budget constraints) establishes the locations of fire stations which minimize the response time in event of a fire. In addition to other applications in facility location, transportation networks, etc. an important application of k -CENTER is in clustering. With the advent of massive data sets, the problem of efficiently and effectively summarizing this data is crucial. A standard approach for this is via centroid-based clustering algorithms of which k -CENTER is a special case. Clustering using k -CENTER has found applications in text summarization, robotics, bioinformatics, pattern recognition, etc. [41, 20, 23, 30].

1.1 Prior work on exact & approximate algorithms for discrete k -Center

The discrete³ k -CENTER problem is NP-hard [44], and admits a 2-approximation [24, 21] in $n^{O(1)}$ time where n is the number of points. This approximation ratio is tight and the k -CENTER problem is NP-hard to approximate in polynomial time to a factor $(2 - \epsilon)$ for any constant $\epsilon > 0$ [25, 21]. Given this intractability, research was aimed at designing parameterized algorithms [10] and parameterized approximation algorithms for k -center. The k -CENTER problem is W[2]-hard to approximate to factor better than 2 even when allowing running times of the form $f(k) \cdot n^{O(1)}$ for any computable function f [15, 13]. The k -CENTER problem remains W[2]-hard even if we combine the parameter k with other structural parameters such as size of vertex cover or size of feedback vertex set [31]. Agarwal and Procopiuc [2] designed an algorithm for discrete k -CENTER on n points in d -dimensional Euclidean space which runs in $n^{O(d \cdot k^{1-1/d})}$ time.

The paradigm of combining parameterized algorithms & approximation algorithms has been successful in designing algorithms for k -center in special topologies such as d -dimensional Euclidean space [2], planar graphs [19], metrics of bounded doubling dimensions [16], graphs of bounded highway dimension [15, 4], etc. Of particular relevance to this paper is the $(1 + \epsilon)$ -approximation algorithm⁴ of Agarwal and Procopiuc [2] which runs in $O(dn \log k) + \left(\frac{k}{\epsilon}\right)^{O(k^{1-1/d})} \cdot n^{O(1)}$ time. This was generalized by Feldmann and Marx [16] who designed an $(1 + \epsilon)$ -approximation algorithm running in $\left(\frac{k^k}{\epsilon^{O(kD)}}\right) \cdot n^{O(1)}$ time for discrete k -CENTER in metric spaces of doubling dimension D .

1.2 From 2-dimensions to higher dimensions

Square root phenomenon for planar graphs and geometric problems in the plane. For a wide range of problems on planar graphs or geometric problems in the plane, a certain *square root phenomenon* is observed for a wide range of algorithmic problems: the exponent of the

³ Here we mention the known results only for the discrete version of k -CENTER. A discussion about results for the continuous version of the problem is given in Section 1.4.

⁴ This is also known as an efficient parameterized approximation scheme (EPAS) as the running time is a function of the type $f(k, \epsilon, d) \cdot n^{O(1)}$.

running time can be improved from $O(\ell)$ to $O(\sqrt{\ell})$ where ℓ is the parameter, or from $O(n)$ to $O(\sqrt{n})$ where n is in the input size, and lower bounds indicate that this improvement is essentially best possible. There is an ever increasing list of such problems known for planar graphs [8, 37, 32, 38, 33, 14, 42, 39, 34, 1, 18] and in the plane [39, 36, 17, 3, 43, 27, 26]

Bounds for higher dimensional Euclidean spaces. Unlike the situation on planar graphs and in two-dimensions, the program of obtaining tight bounds for higher dimensions is still quite nascent with relatively fewer results [9, 40, 5, 12, 11]. Marx and Sidiropoulos [40] showed that for some problems there is a *limited blessing of low dimensionality*: that is, for d -dimensions the running time can be improved from n^ℓ to $n^{\ell^{1-1/d}}$ or from 2^n to $2^{n^{1-1/d}}$ where ℓ is a parameter and n is the input size. In contrast, Cohen-Addad et al. [9] showed that the two problems of k -MEDIAN and k -MEANS suffer from the *curse of low dimensionality*: even for 4-dimensional Euclidean space, assuming the Exponential Time Hypothesis⁵ (ETH), there is no $f(k) \cdot n^{o(k)}$ time algorithm, i.e., the brute force algorithm which runs in $n^{O(k)}$ time is asymptotically optimal.

1.3 Motivation & Our Results

In two-dimensional Euclidean space there is an $n^{O(\sqrt{k})}$ algorithm [2, 27, 26], and a matching lower bound of $f(k) \cdot n^{o(\sqrt{k})}$ under Exponential Time Hypothesis (ETH) for any computable function f [36]. Our motivation in this paper is to investigate what is the *correct* complexity of exact and approximate algorithms for the discrete k -CENTER for higher dimensional Euclidean spaces. In particular, we aim to answer the following two questions:

- (Question 1)** Can the running time of the $(1 + \epsilon)$ -approximation algorithm of [2] be improved from $O(dn \log k) + \left(\frac{k}{\epsilon}\right)^{O(k^{1-1/d})} \cdot n^{O(1)}$, or is there a (close to) matching lower bound?
- (Question 2)** The $n^{O(d \cdot k^{1-1/d})}$ algorithm of [2] for d -dimensional Euclidean space shows that there is a *limited blessing of low dimensionality* for k -CENTER. But can the term $k^{1-1/d}$ in the exponent be improved, or is it asymptotically tight?

We make progress towards answering both these questions by showing the following theorem:

► **Theorem 1.** *For any $d \geq 2$, under the Exponential Time Hypothesis (ETH), the discrete k -CENTER problem in d -dimensional Euclidean space*

- **(Inapproximability result)** *does not admit an $(1 + \epsilon)$ -approximation in $f(k) \cdot \left(\frac{1}{\epsilon}\right)^{o(k^{1-1/d})} \cdot n^{o(k^{1-1/d})}$ time where f is any computable function and n is the number of points.*
- **(Lower bound for exact algorithm)** *cannot be solved in $f(k) \cdot n^{o(k^{1-1/d})}$ time where f is any computable function and n is the number of points.*

Theorem 1 answers Question 1 by showing that the running time of the $(1 + \epsilon)$ -approximation algorithm of Agarwal and Procopiu [2] is essentially tight, i.e., the dependence on ϵ cannot be improved even if we allow a larger dependence on both k and n . Theorem 1 answers Question 2 by showing that the running time of the exact algorithm of Agarwal and Procopiu [2] is asymptotically tight, i.e., the exponent of $k^{1-1/d}$ cannot be asymptotically improved even if we allow a larger dependence on k .

⁵ Recall that the Exponential Time Hypothesis (ETH) has the consequence that n -variable 3-SAT cannot be solved in $2^{o(n)}$ time [28, 29].

1.4 Discussion of the continuous k -Center problem

In the continuous version of the k -CENTER problem, the centers are not required to be picked from the original set of input points. The $n^{O(d \cdot k^{1-1/d})}$ algorithm of Agarwal and Procopiuc [2] also works for this continuous version of the k -CENTER problem in \mathbb{R}^d . Marx [35] showed the W[1]-hardness of k -CENTER in $(\mathbb{R}^2, \ell_\infty)$ parameterized by k . Cabello et al. [6] studied the complexity of this problem parameterized by the dimension, and showed the W[1]-hardness of 4-CENTER in $(\mathbb{R}^d, \ell_\infty)$ parameterized by d . Additionally, they also obtained the W[1]-hardness of 2-CENTER in (\mathbb{R}^d, ℓ_2) parameterized by d ; this reduction also rules out existence of $n^{o(d)}$ algorithms for this problem under the Exponential Time Hypothesis (ETH). It is an interesting open question whether the $n^{O(d \cdot k^{1-1/d})}$ algorithm of Agarwal and Procopiuc [2] is also asymptotically tight for the continuous version of the problem: one way to possibly prove this would be to extend the W[1]-hardness reduction of Marx [35] for continuous k -CENTER in \mathbb{R}^2 (parameterized by k) to higher dimensions using the framework of Marx and Sidiropoulos [40]. Our reduction in this paper does not extend to the continuous version.

1.5 Notation

The set $\{1, 2, \dots, n\}$ is denoted by $[n]$. All vectors considered in this paper have length d . If \mathbf{a} is a vector then for each $i \in [d]$ its i -th coordinate is denoted by $\mathbf{a}[i]$. Addition and subtraction of vectors is denoted by \oplus and \ominus respectively. The i -th unit vector is denoted by \mathbf{e}_i and has $\mathbf{e}_i[i] = 1$ and $\mathbf{e}_i[j] = 0$ for each $j \neq i$. The d -dimensional vector whose every coordinate equals 1 is denoted by $\mathbf{1}^d$. If u is a point and X is a set of points then $\text{dist}(u, X) = \min_{x \in X} \text{dist}(u, x)$. We will sometimes abuse notation slightly and use x to denote both the name and location of the point x .

2 Lower bounds for exact & approximate k -Center in d -dimensional Euclidean space

The goal of this section is to prove Theorem 1 which is restated below:

► **Theorem 1.** *For any $d \geq 2$, under the Exponential Time Hypothesis (ETH), the discrete k -CENTER problem in d -dimensional Euclidean space*

- **(Inapproximability result)** *does not admit an $(1 + \epsilon)$ -approximation in $f(k) \cdot \left(\frac{1}{\epsilon}\right)^{o(k^{1-1/d})} \cdot n^{o(k^{1-1/d})}$ time where f is any computable function and n is the number of points.*
- **(Lower bound for exact algorithm)** *cannot be solved in $f(k) \cdot n^{o(k^{1-1/d})}$ time where f is any computable function and n is the number of points.*

Roadmap to prove Theorem 1. To prove Theorem 1, we design a gap reduction (described in Section 2.2) from a constraint satisfaction problem (CSP) to the k -CENTER problem. The definition and statement of the lower bound for the CSP due to Marx and Sidiropoulos [40] is given in Section 2.1. The correctness of the reduction is shown in Section 2.4 and Section 2.3. Finally, everything is tied together in Section 2.5 which contains the proof of Theorem 1.

2.1 Lower bound for d -dimensional geometric \geq -CSP [40]

This section introduces the d -dimensional geometric \geq -CSP problem of Marx and Sidiropoulos [40]. First we start with some definitions before stating the formal lower bound (Theorem 5) that will be used to prove Theorem 1. Constraint Satisfaction Problems (CSPs) are a general way to represent several important problems in theoretical computer science. In this paper, we will only need a subclass of CSPs called binary CSPs which we define below.

► **Definition 2.** An instance of a binary constraint satisfaction problem (CSP) is a triple $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ where \mathcal{V} is a set of variables, \mathcal{D} is a domain of values and \mathcal{C} is a set of constraints. There are two types of constraints:

- **Unary constraints:** For some $v \in \mathcal{V}$ there is a unary constraint $\langle v, R_v \rangle$ where $R_v \subseteq \mathcal{D}$.
- **Binary constraints:** For some $u, v \in \mathcal{V}$, $u \neq v$, there is a binary constraint $\langle (u, v), R_{u,v} \rangle$ where $R_{u,v} \subseteq \mathcal{D} \times \mathcal{D}$.

Solving a given CSP instance $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ is to check whether there exists a satisfying assignment for it, i.e., a function $f : \mathcal{V} \rightarrow \mathcal{D}$ such that all the constraints are satisfied. For a binary CSP, a satisfying assignment f has the property that for each unary constraint $\langle v, R_v \rangle$ we have $f(v) \in R_v$ and for each binary constraint $\langle (u, v), R_{u,v} \rangle$ we have $(f(u), f(v)) \in R_{u,v}$.

The constraint graph of a given CSP instance $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ is an undirected graph $G_{\mathcal{I}}$ whose vertex set is V and the adjacency relation is defined as follows: two vertices $u, v \in V$ are adjacent in $G_{\mathcal{I}}$ if there is a constraint in \mathcal{I} which contains both u and v . Marx and Sidiropoulos [40] observed that binary CSPs whose primal graph is a subgraph of the d -dimensional grid is useful in showing lower bounds for geometric problems in d -dimensions.

► **Definition 3.** The d -dimensional grid $R[N, d]$ is an undirected graph with vertex set $[N]^d$ and the adjacency relation is as follows: two vertices (a_1, a_2, \dots, a_d) and (b_1, b_2, \dots, b_d) have an edge between them if and only if $\sum_{i=1}^d |a_i - b_i| = 1$.

► **Definition 4.** A d -dimensional geometric \geq -CSP $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ is a binary CSP whose

- set of variables \mathcal{V} is a subset of $R[N, d]$ for some $N \geq 1$,
- domain is $[\delta]^d$ for some integer $\delta \geq 1$,
- constraint graph $G_{\mathcal{I}}$ is an induced subgraph of $R[N, d]$,
- unary constraints are arbitrary, and
- binary constraints are of the following type: if $\mathbf{a}, \mathbf{a}' \in \mathcal{V}$ such that $\mathbf{a}' = \mathbf{a} \oplus \mathbf{e}_i$ for some $i \in [d]$ then there is a binary constraint $\langle (\mathbf{a}, \mathbf{a}'), R_{\mathbf{a}, \mathbf{a}'} \rangle$ where $R_{\mathbf{a}, \mathbf{a}'} = \{(\mathbf{x}, \mathbf{y}) \in R_{\mathbf{a}} \times R_{\mathbf{a}'} \mid \mathbf{x}[i] \geq \mathbf{y}[i]\}$.

Observe that the set of unary constraints of a d -dimensional geometric \geq -CSP is sufficient to completely define it. The size $|\mathcal{I}|$ of a binary CSP $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ is the combined size of the variables, domain and the constraints. With appropriate preprocessing (e.g., combining different constraints on the same variables) we can assume that $|\mathcal{I}| = (|\mathcal{V}| + |\mathcal{D}|)^{O(1)}$. We now state the result of Marx and Sidiropoulos [40] which gives a lower bound on the complexity of checking whether a given d -dimensional geometric \geq -CSP has a satisfying assignment.

► **Theorem 5** ([40, Theorem 2.10]). *If for some fixed $d \geq 2$, there is an $f(|\mathcal{V}|) \cdot |\mathcal{I}|^{o(|\mathcal{V}|^{1-1/d})}$ time algorithm for solving a d -dimensional geometric \geq -CSP \mathcal{I} for some computable function f , then the Exponential Time Hypothesis (ETH) fails.*

► **Remark 6.** The problem defined by Marx and Sidiropoulos [40] is actually d -dimensional geometric \leq -CSP which has \leq -constraints instead of the \geq -constraints. However, for each $\mathbf{a} \in \mathcal{V}$ by replacing each unary constraint $\mathbf{x} \in R_{\mathbf{a}}$ by \mathbf{y} such that $\mathbf{y}[i] = N + 1 - \mathbf{x}[i]$ for each $i \in [d]$, it is easy to see that d -dimensional geometric \leq -CSP and d -dimensional geometric \geq -CSP are equivalent.

2.2 Reduction from d -dimensional geometric \geq -CSP to k -Center in \mathbb{R}^d

We are now ready to describe our reduction from d -dimensional geometric \geq -CSP to k -CENTER in \mathbb{R}^d . Fix any $d \geq 2$. Let $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ be a d -dimensional geometric \geq -CSP instance on variables \mathcal{V} and domain $[\delta]^d$ for some integer $\delta \geq 1$. We fix⁶ the following two quantities:

$$r := \frac{1}{4} \quad \text{and} \quad \epsilon := \frac{r^2}{(d-1)\delta^2} = \frac{1}{16(d-1)\delta^2}. \quad (1)$$

Since $d \geq 2$ and $\delta \geq 1$, we obtain the following bounds from Equation 1,

$$0 < \epsilon \leq \epsilon\delta \leq \epsilon\delta^2 \leq \epsilon\delta^2(d-1) = r^2 = \frac{1}{16}. \quad (2)$$

Given an instance $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ of d -dimensional geometric \geq -CSP, we add a set \mathcal{U} of points in \mathbb{R}^d as described in Table 1 and Table 2. These set of points are the input for the instance of the $|\mathcal{V}|$ -CENTER problem.

■ **Table 1** The set \mathcal{U} of points in \mathbb{R}^d (which gives an instance of k -CENTER) constructed from an instance $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ of d -dimensional geometric \geq -CSP.

- (1) Corresponding to variables: If $\mathbf{a} \in \mathcal{V}$ then we add the following set of points which are collectively called as $\text{BORDER}[\mathbf{a}]$
 - For each $i \in [d]$, the point $B_{\mathbf{a}}^{+i}$ which is located at $\mathbf{a} \oplus \mathbf{e}_i \cdot r(1 - \epsilon) \oplus (\mathbf{1}^d - \mathbf{e}_i) \cdot 2\epsilon\delta$.
 - For each $i \in [d]$, the point $B_{\mathbf{a}}^{-i}$ which is located at $\mathbf{a} \ominus \mathbf{e}_i \cdot r(1 - \epsilon) \ominus (\mathbf{1}^d - \mathbf{e}_i) \cdot 2\epsilon\delta$.

This set of points are referred to as *border* points.
- (2) Corresponding to unary constraints: If $\mathbf{a} \in \mathcal{V}$ and $\langle \mathbf{a}, R_{\mathbf{a}} \rangle$ is the unary constraint on \mathbf{a} , then we add the following set of points which are collectively called as $\text{CORE}[\mathbf{a}]$:
 - for each $\mathbf{x} \in R_{\mathbf{a}} \subseteq [\delta]^d$ we add a point called $C_{\mathbf{a}}^{\mathbf{x}}$ located at $\mathbf{a} \oplus \epsilon \cdot \mathbf{x}$.

This set of points are referred to as *core* points.
- (3) Corresponding to adjacencies in $G_{\mathcal{I}}$: For every edge $(\mathbf{a}, \mathbf{a}')$ in $G_{\mathcal{I}}$ we add a collection of δ points denoted by $\mathcal{S}_{\{\mathbf{a}, \mathbf{a}'\}}$. Assume, without loss of generality, that $\mathbf{a}' = \mathbf{a} \oplus \mathbf{e}_i$ for some $i \in [d]$. Then the set of points $\mathcal{S}_{\{\mathbf{a}, \mathbf{a}'\}}$ is defined as follows:
 - for each $\ell \in [\delta]$ we add a point $S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell}$ which is located at $\mathbf{a} \oplus \mathbf{e}_i \cdot ((1 - \epsilon)2r + \epsilon\ell)$.

This set of points are referred to as *secondary* points.

Note that we add at most $|\mathcal{V}| \cdot 2d$ many border points, at most $|\mathcal{C}|$ many core points, and at most $|\mathcal{V}|^2 \cdot \delta$ many secondary points. Hence, the total number of points n in the instance \mathcal{U} is $\leq |\mathcal{V}| \cdot 2d + |\mathcal{C}| + |\mathcal{V}|^2 \cdot \delta = |\mathcal{I}|^{O(1)}$ where $|\mathcal{I}| = |\mathcal{V}| + |\mathcal{D}| + |\mathcal{C}|$. We now prove some preliminary lemmas to be later used in Section 2.4 and Section 2.3.

2.2.1 Preliminary lemmas

► **Lemma 7.** For each $\mathbf{a} \in \mathcal{V}$ and $i \in [d]$, we have $\text{dist}(B_{\mathbf{a}}^{+i}, B_{\mathbf{a}}^{-i}) \geq 2r(1 + \epsilon)$.

⁶ For simplicity of presentation, we choose $r = 1/4$ instead of $r = 1$: by scaling the result holds for $r = 1$.

■ **Table 2** Notation for some special subsets of points from \mathcal{U} . Note that a primary point is either a core point or a border point.

For each $\mathbf{a} \in \mathcal{V}$, let $\mathcal{D}[\mathbf{a}] := \text{CORE}[\mathbf{a}] \cup \text{BORDER}[\mathbf{a}]$.	(3)
The set of primary points is $\text{PRIMARY} := \bigcup_{\mathbf{a} \in \mathcal{V}} \mathcal{D}[\mathbf{a}]$.	(4)
The set of secondary points is $\text{SECONDARY} := \bigcup_{\mathbf{a} \ \& \ \mathbf{a}' \text{ forms an edge in } G_{\mathcal{I}}} \mathcal{S}_{\{\mathbf{a}, \mathbf{a}'\}}$.	(5)
The final collection of points is $\mathcal{U} := \text{PRIMARY} \cup \text{SECONDARY}$.	(6)

Proof. Fix any $\mathbf{a} \in \mathcal{V}$ and $i \in [d]$. By Table 1, the points $B_{\mathbf{a}}^{+i}$ and $B_{\mathbf{a}}^{-i}$ are located at $\mathbf{a} \oplus \mathbf{e}_i \cdot r(1 - \epsilon) \oplus (\mathbf{1}^d - \mathbf{e}_i) \cdot 2\epsilon\delta$ and $\mathbf{a} \ominus \mathbf{e}_i \cdot r(1 - \epsilon) \ominus (\mathbf{1}^d - \mathbf{e}_i) \cdot 2\epsilon\delta$ respectively. Hence, we have that

$$\begin{aligned} \text{dist}(B_{\mathbf{a}}^{+i}, B_{\mathbf{a}}^{-i})^2 &= (2r(1 - \epsilon))^2 + (d - 1) \cdot (4\epsilon\delta)^2 = (2r(1 - \epsilon))^2 + 16\epsilon \cdot (d - 1)\epsilon\delta^2, \\ &= (2r(1 - \epsilon))^2 + 16\epsilon \cdot r^2, && \text{(by definition of } \epsilon \text{ in Equation 1)} \\ &= (2r)^2[(1 - \epsilon)^2 + 4\epsilon] = (2r(1 + \epsilon))^2. \end{aligned}$$

◀

► **Lemma 8.** For each $\mathbf{a} \in \mathcal{V}$, the distance between any two points in $\text{CORE}[\mathbf{a}]$ is $< r$.

Proof. Fix any $\mathbf{a} \in \mathcal{V}$. Consider any two points in $\text{CORE}[\mathbf{a}]$, say $C_{\mathbf{a}}^{\mathbf{x}}$ and $C_{\mathbf{a}}^{\mathbf{y}}$, for some $\mathbf{x} \neq \mathbf{y}$. By Table 1, these points are located at $\mathbf{a} \oplus \epsilon \cdot \mathbf{x}$ and $\mathbf{a} \oplus \epsilon \cdot \mathbf{y}$ respectively. Hence, we have

$$\begin{aligned} \text{dist}(C_{\mathbf{a}}^{\mathbf{x}}, C_{\mathbf{a}}^{\mathbf{y}})^2 &= (\epsilon \cdot \text{dist}(\mathbf{x}, \mathbf{y}))^2, \\ &\leq \epsilon^2 \cdot d \cdot (\delta - 1)^2, && \text{(since } \mathbf{x}, \mathbf{y} \in R_{\mathbf{a}} \subseteq [\delta]^d) \\ &= \frac{d(\delta - 1)^2}{(d - 1)^2\delta^4} \cdot r^4, && \text{(by definition of } \epsilon \text{ in Equation 1)} \\ &\leq \frac{1}{8} \cdot r^4 < r. && \text{(since } d \geq 2 \text{ and } \delta \geq 1) \end{aligned}$$

◀

► **Lemma 9.** For each $\mathbf{a} \in \mathcal{V}$, the distance of any point from $\text{CORE}[\mathbf{a}]$ to any point from $\text{BORDER}[\mathbf{a}]$ is $< 2r$.

Proof. Fix any $\mathbf{a} \in \mathcal{V}$ and consider any point $C_{\mathbf{a}}^{\mathbf{x}} \in \text{CORE}[\mathbf{a}]$ where $\mathbf{x} \in R_{\mathbf{a}} \subseteq [\delta]^d$. We prove this lemma by showing that, for each $i \in [d]$, the point $C_{\mathbf{a}}^{\mathbf{x}}$ is at distance $< 2r$ from both the points $B_{\mathbf{a}}^{+i}$ and $B_{\mathbf{a}}^{-i}$. Fix some $i \in [d]$.

(i) By Table 1, the points $C_{\mathbf{a}}^{\mathbf{x}}$ and $B_{\mathbf{a}}^{+i}$ are located at $\mathbf{a} \oplus \epsilon \cdot \mathbf{x}$ and $\mathbf{a} \oplus \mathbf{e}_i \cdot r(1 - \epsilon) \oplus (\mathbf{1}^d - \mathbf{e}_i) \cdot 2\epsilon\delta$ respectively. Hence, we have

$$\begin{aligned} \text{dist}(C_{\mathbf{a}}^{\mathbf{x}}, B_{\mathbf{a}}^{+i})^2 &= (r(1 - \epsilon) - \epsilon \cdot \mathbf{x}[i])^2 + \sum_{j=1: j \neq i}^d (2\epsilon\delta - \epsilon \cdot \mathbf{x}[j])^2, \\ &\leq (r(1 - \epsilon))^2 + (d - 1)(2\epsilon\delta)^2, && \text{(since } \mathbf{x}[i], \mathbf{x}[j] \geq 1) \\ &= (r(1 - \epsilon))^2 + 4\epsilon r^2, && \text{(by definition of } \epsilon \text{ in Equation 1)} \\ &= (r(1 + \epsilon))^2 < (2r)^2. && \text{(since } \epsilon < 1) \end{aligned}$$

28:8 Tight Lower Bounds for Approximate & Exact k -Center in \mathbb{R}^d

(ii) By Table 1, the points $C_{\mathbf{a}}^{\mathbf{x}}$ and $B_{\mathbf{a}}^{-i}$ are located at $\mathbf{a} \oplus \epsilon \cdot \mathbf{x}$ and $\mathbf{a} \ominus \mathbf{e}_i \cdot r(1-\epsilon) \ominus (\mathbf{1}^d - \mathbf{e}_i) \cdot 2\epsilon\delta$ respectively. Hence, we have

$$\begin{aligned} \text{dist}(C_{\mathbf{a}}^{\mathbf{x}}, B_{\mathbf{a}}^{-i})^2 &= (r(1-\epsilon) + \epsilon \cdot \mathbf{x}[i])^2 + \sum_{j=1: j \neq i}^d (\epsilon \cdot \mathbf{x}[j] + 2\epsilon\delta)^2, \\ &\leq (r(1-\epsilon) + \epsilon\delta)^2 + (d-1)(3\epsilon\delta)^2, && \text{(since } \mathbf{x}[i], \mathbf{x}[j] \leq \delta) \\ &= (r(1-\epsilon) + \epsilon\delta)^2 + 9\epsilon r^2, && \text{(by definition of } \epsilon) \\ &\leq 2r^2(1-\epsilon)^2 + 2\epsilon^2\delta^2 + 9\epsilon r^2, && \text{(since } (\alpha + \beta)^2 \leq 2\alpha^2 + 2\beta^2) \\ &\leq 2r^2(1-\epsilon)^2 + 11\epsilon r^2, && \text{(since } \epsilon\delta^2 \leq r^2) \\ &= 2r^2((1-\epsilon)^2 + 5.5\epsilon) < 2r^2(1 + 1.75\epsilon)^2 < (2r)^2. && \text{(since } \epsilon \leq 1/16) \end{aligned}$$

◀

► **Lemma 10.** For each $\mathbf{a} \in \mathcal{V}$, the distance of \mathbf{a} to any point in $\text{BORDER}[\mathbf{a}]$ is $r(1+\epsilon)$.

Proof. Let p be any point in $\text{BORDER}[\mathbf{a}]$. Then we have two choices for p , namely $p = B_{\mathbf{a}}^{+i}$ or $p = B_{\mathbf{a}}^{-i}$. In both cases, we have

$$\text{dist}(p, \mathbf{a})^2 = (r(1-\epsilon))^2 + (d-1)(2\epsilon\delta)^2 = r^2(1-\epsilon)^2 + 4\epsilon r^2 = (r(1+\epsilon))^2,$$

where the second equality is obtained by the definition of ϵ (Equation 1). ◀

► **Lemma 11.** For each $\mathbf{a} \in \mathcal{V}$ and each $i \in [d]$,

- If $w \in \mathcal{U}$ such that $\text{dist}(w, B_{\mathbf{a}}^{+i}) < 2r(1+\epsilon)$ then $w \in (\mathcal{D}[\mathbf{a}] \cup \mathcal{S}_{\{\mathbf{a}, \mathbf{a} \oplus \mathbf{e}_i\}})$.
- If $w \in \mathcal{U}$ such that $\text{dist}(w, B_{\mathbf{a}}^{-i}) < 2r(1+\epsilon)$ then $w \in (\mathcal{D}[\mathbf{a}] \cup \mathcal{S}_{\{\mathbf{a}, \mathbf{a} \ominus \mathbf{e}_i\}})$.

Proof. The proof of this lemma is deferred to the full version [7]. ◀

► **Remark 12.** Lemma 11 gives a necessary but not sufficient condition. Also, it might be the case that for some $\mathbf{a} \in \mathcal{V}$ and $i \in [d]$ the vector $\mathbf{a} \oplus \mathbf{e}_i \notin \mathcal{V}$ (resp., $\mathbf{a} \ominus \mathbf{e}_i \notin \mathcal{V}$) in which case the set $\mathcal{S}_{\{\mathbf{a}, \mathbf{a} \oplus \mathbf{e}_i\}}$ (resp., $\mathcal{S}_{\{\mathbf{a}, \mathbf{a} \ominus \mathbf{e}_i\}}$) is empty.

► **Lemma 13.** Let $\mathbf{a} \in \mathcal{V}$ and $i \in [d]$ be such that $\mathbf{a}' := (\mathbf{a} \oplus \mathbf{e}_i) \in \mathcal{V}$. For each $\ell \in [\delta]$,

- (1) If $\mathbf{x} \in R_{\mathbf{a}}$ and $\ell \leq \mathbf{x}[i]$, then $\text{dist}(C_{\mathbf{a}}^{\mathbf{x}}, S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell}) < 2r$.
- (2) If $\mathbf{x} \in R_{\mathbf{a}}$ and $\ell > \mathbf{x}[i]$, then $\text{dist}(C_{\mathbf{a}}^{\mathbf{x}}, S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell}) \geq 2r(1+\epsilon)$.
- (3) If $\mathbf{y} \in R_{\mathbf{a}'}$ and $\ell > \mathbf{y}[i]$, then $\text{dist}(C_{\mathbf{a}'}^{\mathbf{y}}, S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell}) < 2r$.
- (4) If $\mathbf{y} \in R_{\mathbf{a}'}$ and $\ell \leq \mathbf{y}[i]$, then $\text{dist}(C_{\mathbf{a}'}^{\mathbf{y}}, S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell}) \geq 2r(1+\epsilon)$.

Proof. Recall from Table 1 that the points $C_{\mathbf{a}}^{\mathbf{x}}$ and $S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell}$ are located at $\mathbf{a} \oplus \epsilon \cdot \mathbf{x}$ and $\mathbf{a} \oplus \mathbf{e}_i \cdot ((1-\epsilon)2r + \epsilon\ell)$ respectively.

$$\begin{aligned} \text{(1) If } \ell \leq \mathbf{x}[i], \text{ then } \text{dist}(C_{\mathbf{a}}^{\mathbf{x}}, S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell})^2 &= (2r(1-\epsilon) + \epsilon(\ell - \mathbf{x}[i]))^2 + \sum_{j=1: j \neq i}^d (\epsilon \cdot \mathbf{x}[j])^2, \\ &\leq (2r(1-\epsilon))^2 + (d-1)\epsilon^2\delta^2 = (2r(1-\epsilon))^2 + \epsilon r^2 && \text{(since } \ell \leq \mathbf{x}[i] \text{ and } \mathbf{x}[j] \leq \delta) \\ &= (2r)^2 \left((1-\epsilon)^2 + \frac{\epsilon}{4} \right) < (2r)^2. && \text{(since } 0 < \epsilon < 1) \end{aligned}$$

(2) If $\ell > \mathbf{x}[i]$, then $\text{dist} \left(C_{\mathbf{a}}^{\mathbf{x}}, S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell} \right)^2$

$$= (2r(1 - \epsilon) + \epsilon(\ell - \mathbf{x}[i]))^2 + \sum_{j=1: j \neq i}^d (\epsilon \cdot \mathbf{x}[j])^2,$$

$$\geq (2r(1 - \epsilon) + \epsilon)^2 = (2r(1 - \epsilon) + 4r\epsilon)^2 = (2r(1 + \epsilon))^2. \quad (\text{since } \ell > \mathbf{x}[i] \text{ and } 4r = 1)$$

We now show the remaining two claims: recall from Table 1 that the points $C_{\mathbf{a}}^{\mathbf{y}}$ and $S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell}$ are located at $(\mathbf{a}' \oplus \epsilon \cdot \mathbf{y}) = \mathbf{a} \oplus \mathbf{e}_i \oplus \epsilon \cdot \mathbf{y}$ and $\mathbf{a} \oplus \mathbf{e}_i \cdot ((1 - \epsilon)2r + \epsilon\ell)$ respectively.

(3) If $\ell > \mathbf{y}[i]$, then $\text{dist} \left(C_{\mathbf{a}'}^{\mathbf{y}}, S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell} \right)^2$

$$= (1 + \epsilon \cdot \mathbf{y}[i] - (1 - \epsilon)2r - \epsilon\ell)^2 + \sum_{j=1: j \neq i}^d (\epsilon \cdot \mathbf{y}[j])^2,$$

$$\leq (4r + \epsilon \cdot \mathbf{y}[i] - (1 - \epsilon)2r - \epsilon\ell)^2 + (d - 1)\epsilon^2\delta^2, \quad (\text{since } 4r = 1 \text{ and } \mathbf{y}[j] \leq \delta)$$

$$= (2r(1 + \epsilon) - \epsilon(\ell - \mathbf{y}[i]))^2 + \epsilon r^2, \quad (\text{since } (d - 1)\epsilon\delta^2 = r^2)$$

$$\leq (2r(1 + \epsilon) - \epsilon)^2 + \epsilon r^2, \quad (\text{since } \ell > \mathbf{y}[i])$$

$$= (2r(1 - \epsilon))^2 + \epsilon r^2, \quad (\text{since } 4r = 1)$$

$$= (2r)^2 \left((1 - \epsilon)^2 + \frac{\epsilon}{4} \right) < (2r)^2. \quad (\text{since } 0 < \epsilon < 1)$$

(4) If $\ell \leq \mathbf{y}[i]$, then $\text{dist} \left(C_{\mathbf{a}'}^{\mathbf{y}}, S_{\{\mathbf{a}, \mathbf{a}'\}}^{\ell} \right)^2$

$$= (1 + \epsilon \cdot \mathbf{y}[i] - (1 - \epsilon)2r - \epsilon\ell)^2 + \sum_{j=1: j \neq i}^d (\epsilon \cdot \mathbf{y}[j])^2,$$

$$\geq (2r(1 + \epsilon) + \epsilon(\mathbf{y}[i] - \ell))^2, \quad (\text{since } 4r = 1)$$

$$\geq (2r(1 + \epsilon))^2. \quad (\text{since } \mathbf{y}[i] \geq \ell)$$



► **Lemma 14.** Let $\mathbf{a} \in \mathcal{V}$ and $i \in [d]$ be such that $\mathbf{a}' := (\mathbf{a} \oplus \mathbf{e}_i) \in \mathcal{V}$. If $\mathbf{a}'' \notin \{\mathbf{a}, \mathbf{a}'\}$ then the distance between any point in $\text{CORE}[\mathbf{a}'']$ and any point in $\mathcal{S}_{\mathbf{a}, \mathbf{a}'}$ is at least $2r(1 + \epsilon)$.

Proof. Let \mathbf{p} and \mathbf{q} be two arbitrary points from $\text{CORE}[\mathbf{a}'']$ and $\mathcal{S}_{\mathbf{a}, \mathbf{a}'}$, respectively. By Table 1, \mathbf{p} is located at $\mathbf{a}'' \oplus \epsilon \cdot \mathbf{x}$ for some $\mathbf{x} \in R_{\mathbf{a}} \subseteq [\delta]^d$ and \mathbf{q} is located at $\mathbf{a} \oplus \mathbf{e}_i \cdot ((1 - \epsilon)2r + \epsilon\ell)$ for some $\ell \in [\delta]$.

Since $\mathbf{a}' = \mathbf{a} \oplus \mathbf{e}_i$ and $\mathbf{a}'' \notin \{\mathbf{a}, \mathbf{a}'\}$, we have three cases to consider:

- $\mathbf{a}''[j] = \mathbf{a}[j]$ for all $j \neq i$ and $\mathbf{a}''[i] \leq \mathbf{a}[i] - 1$: In this case, we have $\text{dist}(\mathbf{p}, \mathbf{q})^2$

$$\geq ((\mathbf{a}[i] + (1 - \epsilon)2r + \epsilon\ell) - (\mathbf{a}''[i] + \epsilon \cdot \mathbf{x}[i]))^2,$$

(only considering the i -th coordinate)

$$= (\mathbf{a}[i] - \mathbf{a}''[i] + (1 - \epsilon)2r + \epsilon\ell - \epsilon\mathbf{x}[i])^2,$$

$$\geq (1 + (1 - \epsilon)2r + \epsilon \cdot 4r - \epsilon\delta)^2, \quad (\text{since } \mathbf{a}[i] - \mathbf{a}''[i] \geq 1, \ell \geq 1 = 4r \text{ and } \mathbf{x}[i] \leq \delta)$$

$$> (2r(1 + \epsilon))^2. \quad (\text{since } 1 - \epsilon\delta \geq 1 - \frac{1}{16} > 0)$$

28:10 Tight Lower Bounds for Approximate & Exact k -Center in \mathbb{R}^d

- $\mathbf{a}''[j] = \mathbf{a}[j]$ for all $j \neq i$ and $\mathbf{a}''[i] \geq \mathbf{a}[i] + 2$: In this case, we have $\text{dist}(\mathbf{p}, \mathbf{q})^2$

$$\begin{aligned} &\geq ((\mathbf{a}''[i] + \epsilon \cdot \mathbf{x}[i]) - (\mathbf{a}[i] + (1 - \epsilon)2r + \epsilon\ell))^2, && \text{(only considering the } i\text{-th coordinate)} \\ &= (\mathbf{a}''[i] - \mathbf{a}[i] - (1 - \epsilon)2r + \epsilon \cdot \mathbf{x}[i] - \epsilon\ell)^2, \\ &\geq (2 - (1 - \epsilon)2r + \epsilon - \epsilon\delta)^2, && \text{(since } \mathbf{a}''[i] - \mathbf{a}[i] \geq 2, \mathbf{x}[i] \geq 1 \text{ and } \ell \leq \delta) \\ &= (4r - (1 - \epsilon)2r + 1 + \epsilon - \epsilon\delta)^2, && \text{(since } 4r = 1) \\ &> (2r(1 + \epsilon))^2. && \text{(since } 1 - \epsilon\delta \geq 1 - \frac{1}{16} > 0) \end{aligned}$$
- There exists $j \neq i$ such that $\mathbf{a}''[j] \neq \mathbf{a}[j]$: In this case, we have $\text{dist}(\mathbf{p}, \mathbf{q})$

$$\begin{aligned} &\geq |\mathbf{a}[j] - (\mathbf{a}''[j] + \epsilon \cdot \mathbf{x}[j])|, && \text{(only considering the } j\text{-th coordinate)} \\ &\geq |\mathbf{a}[j] - \mathbf{a}''[j]| - \epsilon \cdot \mathbf{x}[j], && \text{(by triangle inequality)} \\ &\geq 1 - \epsilon \cdot \delta, && \text{(since } \mathbf{a}[j] \neq \mathbf{a}''[j] \text{ and } \mathbf{x}[j] \leq \delta) \\ &\geq 2r + 2r - r^2 = 2r + 2r \left(1 - \frac{r}{2}\right), && \text{(since } 4r = 1 \text{ and } \epsilon\delta \leq r^2) \\ &> 2r(1 + \epsilon). && \text{(since } 1 - \frac{r}{2} > \frac{1}{16} \geq \epsilon) \end{aligned}$$

◀

2.3 \mathcal{I} has a satisfying assignment \Rightarrow OPT for the instance \mathcal{U} of $|\mathcal{V}|$ -Center is $< 2r$

Suppose that the d -dimensional geometric \geq -CSP, $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$, has a satisfying assignment $f : \mathcal{V} \rightarrow \mathcal{D}$. Consider the set of points F given by $\{C_{\mathbf{a}}^{f(\mathbf{a})} : \mathbf{a} \in \mathcal{V}\}$. Since $f : \mathcal{V} \rightarrow \mathcal{D}$ is a satisfying assignment for \mathcal{I} , it follows that $f(\mathbf{a}) \in R_{\mathbf{a}}$ for each $\mathbf{a} \in \mathcal{V}$ and hence the set F is well-defined. Clearly, $|F| = |\mathcal{V}|$. We now show that

$$\text{OPT}(F) := \left(\max_{u \in \mathcal{U}} \left(\min_{v \in F} \text{dist}(u, v) \right) \right) < 2r.$$

This implies that OPT for the instance \mathcal{U} of $|\mathcal{V}|$ -CENTER is $< 2r$. We show $\text{OPT}(F) < 2r$ by showing that $\text{dist}(p, F) < 2r$ for each $p \in \mathcal{U}$. From Table 1 and Table 2, it is sufficient to consider the two cases depending on whether p is a primary point or a secondary point.

► **Lemma 15.** *If p is a primary point, then $\text{dist}(p, F) < 2r$.*

Proof. If p is a primary point, then by Table 1 and Table 2 it follows that p is either a core point or a border point.

- **p is a core point:** By Table 1, $p \in \text{CORE}[\mathbf{b}]$ for some $\mathbf{b} \in \mathcal{V}$. Then, Lemma 8 implies that $\text{dist}(p, C_{\mathbf{b}}^{f(\mathbf{b})}) < r$. Since $C_{\mathbf{b}}^{f(\mathbf{b})} \in F$, we have $\text{dist}(p, F) \leq \text{dist}(p, C_{\mathbf{b}}^{f(\mathbf{b})}) < r$.
- **p is a border point:** By Table 1, $p \in \text{BORDER}[\mathbf{b}]$ for some $\mathbf{b} \in \mathcal{V}$. Then, Lemma 9 implies that $\text{dist}(p, C_{\mathbf{b}}^{f(\mathbf{b})}) < 2r$. Since $C_{\mathbf{b}}^{f(\mathbf{b})} \in F$, we have $\text{dist}(p, F) \leq \text{dist}(p, C_{\mathbf{b}}^{f(\mathbf{b})}) < 2r$.

◀

► **Lemma 16.** *If p is a secondary point, then $\text{dist}(p, F) < 2r$.*

Proof. If p is a secondary point, then by Table 1 and Table 2 it follows that there exists $\mathbf{a} \in \mathcal{V}, i \in [d]$ and $\ell \in [\delta]$ such that $p = S_{\{\mathbf{a}, \mathbf{a} \oplus \mathbf{e}_i\}}^\ell$. Note that $C_{\mathbf{a}}^{f(\mathbf{a})} \in F$ and $C_{\mathbf{a} \oplus \mathbf{e}_i}^{f(\mathbf{a} \oplus \mathbf{e}_i)} \in F$.

We now prove the lemma by showing that $\min \left\{ \text{dist}(p, C_{\mathbf{a}}^{f(\mathbf{a})}), \text{dist}(p, C_{\mathbf{a} \oplus \mathbf{e}_i}^{f(\mathbf{a} \oplus \mathbf{e}_i)}) \right\} < 2r$. Since $f : \mathcal{V} \rightarrow \mathcal{D}$ is a satisfying assignment, the binary constraint on \mathbf{a} and $\mathbf{a} \oplus \mathbf{e}_i$ is satisfied, i.e., $\delta \geq f(\mathbf{a})[i] \geq f(\mathbf{a} \oplus \mathbf{e}_i)[i] \geq 1$. Since $\ell \in [\delta]$, either $\ell \leq f(\mathbf{a})[i]$ or $\ell > f(\mathbf{a} \oplus \mathbf{e}_i)[i]$. The following implications complete the proof:

- If $\ell \leq f(\mathbf{a})[i]$, then 13(1) implies that $\text{dist}\left(C_{\mathbf{a}}^{f(\mathbf{a})}, p\right) < 2r$.
- If $\ell > f(\mathbf{a} \oplus \mathbf{e}_i)[i]$, then 13(3) implies that $\text{dist}\left(C_{\mathbf{a} \oplus \mathbf{e}_i}^{f(\mathbf{a} \oplus \mathbf{e}_i)}, p\right) < 2r$. ◀

From Table 2, Lemma 15 and Lemma 16 it follows that OPT for the instance \mathcal{U} of $|\mathcal{V}|$ -CENTER is $< 2r$.

2.4 \mathcal{I} does not have a satisfying assignment \Rightarrow OPT for the instance \mathcal{U} of $|\mathcal{V}|$ -Center is $\geq 2r(1 + \epsilon)$

Suppose that the instance $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ of d -dimensional geometric \geq -CSP does not have a satisfying assignment. We want to now show that OPT for the instance \mathcal{U} of $|\mathcal{V}|$ -CENTER is $\geq 2r(1 + \epsilon)$. Fix any set $Q \subseteq \mathcal{U}$ of size $|\mathcal{V}|$: it is sufficient to show that

$$\text{OPT}(Q) := \left(\max_{u \in \mathcal{U}} \left(\min_{v \in Q} \text{dist}(u, v) \right) \right) \geq 2r(1 + \epsilon). \tag{7}$$

We consider two cases: either $|Q \cap \text{CORE}[\mathbf{a}]| = 1$ for each $\mathbf{a} \in \mathcal{V}$ (Lemma 17) or not (Lemma 18).

▶ **Lemma 17.** *If $|Q \cap \text{CORE}[\mathbf{a}]| = 1$ for each $\mathbf{a} \in \mathcal{V}$ then $\text{OPT}(Q) \geq 2r(1 + \epsilon)$.*

Proof. Since $|Q| = |\mathcal{V}|$ and $|Q \cap \text{CORE}[\mathbf{a}]| = 1$ for each $\mathbf{a} \in \mathcal{V}$ it follows that the only points in Q are core points (see Table 1 for definition) and moreover Q contains exactly one core point corresponding to each element from \mathcal{V} . Let $\phi : \mathcal{V} \rightarrow [\delta]^d$ be the function such that $Q \cap \text{CORE}[\mathbf{a}] = C_{\mathbf{a}}^{\phi(\mathbf{a})}$. By Table 1, it follows that $\phi(\mathbf{a}) \in R_{\mathbf{a}}$ for each $\mathbf{a} \in \mathcal{V}$.

Recall that we are assuming in this section that the instance $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ of d -dimensional geometric \geq -CSP does not have a satisfying assignment. Hence, in particular, the function $\phi : \mathcal{V} \rightarrow [\delta]^d$ is not a satisfying assignment for \mathcal{I} . All unary constraints are satisfied since $\phi(\mathbf{a}) \in R_{\mathbf{a}}$ for each $\mathbf{a} \in \mathcal{V}$. Hence, there is some binary constraint which is not satisfied by ϕ : let this constraint be violated for the pair $\mathbf{a}, \mathbf{a} \oplus \mathbf{e}_i$ for some $\mathbf{a} \in \mathcal{V}$ and $i \in [d]$. Let us denote $\mathbf{a} \oplus \mathbf{e}_i$ by \mathbf{a}' . The violation of the binary constraint on \mathbf{a} and $\mathbf{a} \oplus \mathbf{e}_i$ by ϕ implies that $1 \leq \phi(\mathbf{a})[i] < \phi(\mathbf{a}')[i] \leq \delta$. We now show that $\text{dist}\left(Q, S_{\{\mathbf{a}, \mathbf{a}'\}}^{\phi(\mathbf{a}') [i]}\right) \geq (2r(1 + \epsilon))$ which, in turn, implies that $\text{OPT}(Q) \geq 2r(1 + \epsilon)$. The following implications complete the proof.

- 13(2) implies that $\text{dist}\left(S_{\{\mathbf{a}, \mathbf{a}'\}}^{\phi(\mathbf{a}') [i]}, C_{\mathbf{a}}^{\phi(\mathbf{a})}\right) \geq 2r(1 + \epsilon)$.
- 13(4) implies that $\text{dist}\left(S_{\{\mathbf{a}, \mathbf{a}'\}}^{\phi(\mathbf{a}') [i]}, C_{\mathbf{a}'}^{\phi(\mathbf{a}')}\right) \geq 2r(1 + \epsilon)$.
- Consider any point $s \in Q \setminus \left\{C_{\mathbf{a}}^{\phi(\mathbf{a})}, C_{\mathbf{a}'}^{\phi(\mathbf{a}')}\right\}$. Then $s \in \text{CORE}[\mathbf{a}'']$ for some $\mathbf{a}'' \notin \{\mathbf{a}, \mathbf{a}'\}$.
Now Lemma 14 implies that $\text{dist}\left(S_{\{\mathbf{a}, \mathbf{a}'\}}^{\phi(\mathbf{a}') [i]}, s\right) \geq 2r(1 + \epsilon)$. ◀

▶ **Lemma 18.** *If there exists $\mathbf{a} \in \mathcal{V}$ such that $|Q \cap \text{CORE}[\mathbf{a}]| \neq 1$ then $\text{OPT}(Q) \geq 2r(1 + \epsilon)$.*

Proof. Suppose that $\text{OPT}(Q) < 2r(1 + \epsilon)$. To prove the lemma, we will now show that this implies $|Q \cap \text{CORE}[\mathbf{a}]| = 1$ for each $\mathbf{a} \in \mathcal{V}$. This is done via the following two claims, namely Claim 19 and Claim 20.

▷ **Claim 19.** $|Q \cap \mathcal{D}[\mathbf{a}]| = 1$ for each $\mathbf{a} \in \mathcal{V}$.

Proof. Define three sets I_0, I_1 and $I_{\geq 2}$ as follows:

$$I_0 := \{\mathbf{a} \in \mathcal{V} : |Q \cap \mathcal{D}[\mathbf{a}]| = 0\} \tag{8}$$

$$I_1 := \{\mathbf{a} \in \mathcal{V} : |Q \cap \mathcal{D}[\mathbf{a}]| = 1\} \tag{9}$$

$$I_{\geq 2} := \{\mathbf{a} \in \mathcal{V} : |Q \cap \mathcal{D}[\mathbf{a}]| \geq 2\} \tag{10}$$

28:12 Tight Lower Bounds for Approximate & Exact k -Center in \mathbb{R}^d

By definition, we have

$$|I_0| + |I_1| + |I_{\geq 2}| = |\mathcal{V}| \quad (11)$$

Consider a variable $\mathbf{b} \in I_0$. Since $\text{dist}(Q, B_{\mathbf{b}}^{+i})$ and $\text{dist}(Q, B_{\mathbf{b}}^{-i}) < 2r(1 + \epsilon)$, and $Q \cap \mathcal{D}[\mathbf{b}] = \emptyset$, Lemma 11 implies that for each $i \in [d]$,

- (i) Q must contain a point from $\mathcal{S}_{\{\mathbf{b}, \mathbf{b} \oplus \mathbf{e}_i\}}$, and
- (ii) Q must contain a point from $\mathcal{S}_{\{\mathbf{b}, \mathbf{b} \ominus \mathbf{e}_i\}}$.

Since each secondary point can be “charged” to two variables in \mathcal{V} (for example, the set $\mathcal{S}_{\{\mathbf{b}, \mathbf{b} \oplus \mathbf{e}_i\}}$ corresponds to both \mathbf{b} and $\mathbf{b} \oplus \mathbf{e}_i$), it follows that Q contains $\geq \frac{2d}{2} = d \geq 2$ distinct secondary points corresponding to each variable in I_0 . Therefore, we have

$$\begin{aligned} |I_0| + |I_1| + |I_{\geq 2}| &= |\mathcal{V}| = |Q|, && \text{(from Equation 11)} \\ &\geq |Q \cap \text{PRIMARY}| + |Q \cap \text{SECONDARY}|, && \\ &&& \text{(since PRIMARY} \cap \text{SECONDARY} = \emptyset) \\ &\geq (|I_1| + 2|I_{\geq 2}|) + |Q \cap \text{SECONDARY}|, && \text{(by definition of } I_1 \text{ and } I_{\geq 2}) \\ &\geq (|I_1| + 2|I_{\geq 2}|) + 2|I_0|, && (12) \end{aligned}$$

where the last inequality follows because Q contains at least 2 secondary points corresponding to each variable in I_0 . Hence, we have $|I_0| + |I_1| + |I_{\geq 2}| \geq 2|I_0| + |I_1| + 2|I_{\geq 2}|$ which implies $|I_0| = 0 = |I_{\geq 2}|$. From Equation 11, we get $|I_1| = |\mathcal{V}|$, i.e., $|Q \cap \mathcal{D}[\mathbf{a}]| = 1$ for each $\mathbf{a} \in \mathcal{V}$. This concludes the proof of Claim 19. \triangleleft

Since $|Q| = |\mathcal{V}|$ and $\mathcal{D}[\mathbf{a}] \cap \mathcal{D}[\mathbf{b}] = \emptyset$ for distinct $\mathbf{a}, \mathbf{b} \in \mathcal{V}$, Claim 19 implies that

$$Q \text{ contains no secondary points.} \quad (13)$$

We now prove that Q doesn't contain border points either.

\triangleright **Claim 20.** $|Q \cap \text{CORE}[\mathbf{a}]| = 1$ for each $\mathbf{a} \in \mathcal{V}$

Proof. Fix any $\mathbf{a} \in \mathcal{V}$. From Claim 19, we know that $|Q \cap \mathcal{D}[\mathbf{a}]| = 1$. Suppose that this unique point in $Q \cap \mathcal{D}[\mathbf{a}]$ is from $\text{BORDER}[\mathbf{a}]$. Without loss of generality, let $Q \cap \mathcal{D}[\mathbf{a}] = \{B_{\mathbf{a}}^{+i}\}$ for some $i \in [d]$. Since $\text{OPT}(Q) < 2r(1 + \epsilon)$, it follows that $\text{dist}(Q, B_{\mathbf{a}}^{-i}) < 2r(1 + \epsilon)$. Hence, Lemma 11(2) implies that $Q \cap (\mathcal{D}[\mathbf{a}] \cup \mathcal{S}_{\{\mathbf{a}, \mathbf{a} \oplus \mathbf{e}_i\}}) \neq \emptyset$. Since Q contains no secondary points (Equation 13), we have $Q \cap (\mathcal{D}[\mathbf{a}] \cup \mathcal{S}_{\{\mathbf{a}, \mathbf{a} \oplus \mathbf{e}_i\}}) = Q \cap \mathcal{D}[\mathbf{a}] = \{B_{\mathbf{a}}^{+i}\}$. But from Lemma 7 we know $\text{dist}(B_{\mathbf{a}}^{+i}, B_{\mathbf{a}}^{-i}) \geq 2r(1 + \epsilon)$. We thus obtain a contradiction. This concludes the proof of Claim 20. \triangleleft

Therefore, we have shown that $\text{OPT}(Q) < 2r(1 + \epsilon)$ implies $|Q \cap \text{CORE}[\mathbf{a}]| = 1$ for each $\mathbf{a} \in \mathcal{V}$. This concludes the proof of Lemma 18. \blacktriangleleft

2.5 Finishing the proof of Theorem 1

Finally, we are ready to prove Theorem 1 which is restated below.

\blacktriangleright **Theorem 1.** *For any $d \geq 2$, under the Exponential Time Hypothesis (ETH), the discrete k -CENTER problem in d -dimensional Euclidean space*

- **(Inapproximability result)** *does not admit an $(1 + \epsilon)$ -approximation in $f(k) \cdot \left(\frac{1}{\epsilon}\right)^{o(k^{1-1/d})} \cdot n^{o(k^{1-1/d})}$ time where f is any computable function and n is the number of points.*
- **(Lower bound for exact algorithm)** *cannot be solved in $f(k) \cdot n^{o(k^{1-1/d})}$ time where f is any computable function and n is the number of points.*

Proof. Given an instance $\mathcal{I} = (\mathcal{V}, \mathcal{D}, \mathcal{C})$ of a d -dimensional geometric \geq -CSP, we build an instance \mathcal{U} of $|\mathcal{V}|$ -CENTER in \mathbb{R}^d given by the reduction in Section 2.2. This reduction has the property that

- if \mathcal{I} does not have a satisfying assignment then OPT for the instance \mathcal{U} of $|\mathcal{V}|$ -CENTER is $\geq 2r(1 + \epsilon^*)$ (Section 2.4), and
- if \mathcal{I} has a satisfying assignment then OPT for the instance \mathcal{U} of $|\mathcal{V}|$ -CENTER is $< 2r$ (Section 2.3),

where $r = 1/4$ and $\epsilon^* = \frac{r^2}{(d-1)\delta^2} \geq \frac{1}{16(d-1)|\mathcal{D}|}$, since $|\mathcal{D}| = |\delta|^d \geq \delta^2$. Hence, any algorithm for the $|\mathcal{V}|$ -center problem which has an approximation factor $\leq (1 + \epsilon^*)$ can solve the d -dimensional geometric \geq -CSP. Note that the instance \mathcal{U} of k -CENTER in \mathbb{R}^d has $k = |\mathcal{V}|$ and the number of points $n \leq |\mathcal{V}| \cdot 2d + |\mathcal{C}| + |\mathcal{V}|^2 \cdot \delta = |\mathcal{I}|^{O(1)}$ where $|\mathcal{I}| = |\mathcal{V}| + |\mathcal{D}| + |\mathcal{C}|$. We now derive the two lower bounds claimed in the theorem.

- **(Inapproximability result)** Suppose that there exists $d \geq 2$ such that the k -center on n points in \mathbb{R}^d admits an $(1 + \epsilon)$ -approximation algorithm in $f(k) \cdot \left(\frac{1}{\epsilon}\right)^{o(k^{1-1/d})} \cdot n^{o(k^{1-1/d})}$ time for some computable function f . As argued above, using a $(1 + \epsilon^*)$ -approximation for the k -center problem with $k = |\mathcal{V}|$ and $n = |\mathcal{I}|^{O(1)}$ points can solve the d -dimensional geometric \geq -CSP problem. Recall that $16(d-1)|\mathcal{I}| \geq 16(d-1)|\mathcal{D}| \geq \frac{1}{\epsilon^*}$ since $|\mathcal{I}| = |\mathcal{V}| + |\mathcal{D}| + |\mathcal{C}|$, and hence we have an algorithm for the d -dimensional geometric \geq -CSP problem which runs in time $f(|\mathcal{V}|) \cdot (16d)^{o(k^{1-1/d})} \cdot |\mathcal{I}|^{o(k^{1-1/d})}$ which contradicts Theorem 5.
- **(Lower bound for exact algorithm)** Suppose that there exists $d \geq 2$ such that the k -center on n points in \mathbb{R}^d admits an exact algorithm in $f(k) \cdot n^{o(k^{1-1/d})}$ time for some computable function f . As argued above⁷, solving the k center problem with $k = |\mathcal{V}|$ and $n = |\mathcal{I}|^{O(1)}$ points can solve the d -dimensional geometric \geq -CSP problem. Hence, we have an algorithm for the d -dimensional geometric \geq -CSP problem which runs in time $f(|\mathcal{V}|) \cdot |\mathcal{I}|^{o(k^{1-1/d})}$ which again contradicts Theorem 5. ◀

References

- 1 Pierre Aboulker, Nick Brettell, Frédéric Havet, Dániel Marx, and Nicolas Trotignon. Coloring graphs with constraints on connectivity. *Journal of Graph Theory*, 85(4):814–838, 2017.
- 2 Pankaj K. Agarwal and Cecilia Magdalena Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002.
- 3 Jochen Alber and Jiri Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *J. Algorithms*, 52(2):134–151, 2004.
- 4 Amariah Becker, Philip N. Klein, and David Saulpic. Polynomial-Time Approximation Schemes for k -center, k -median, and Capacitated Vehicle Routing in Bounded Highway Dimension. In *ESA 2018*, volume 112, pages 8:1–8:15, 2018.
- 5 Csaba Biró, Édouard Bonnet, Dániel Marx, Tillmann Miltzow, and Pawel Rzazewski. Fine-grained complexity of coloring unit disks and balls. *J. Comput. Geom.*, 9(2):47–80, 2018. doi:10.20382/jocg.v9i2a4.
- 6 Sergio Cabello, Panos Giannopoulos, Christian Knauer, Dániel Marx, and Günter Rote. Geometric clustering: Fixed-parameter tractability and lower bounds with respect to the dimension. *ACM Trans. Algorithms*, 7(4):43:1–43:27, 2011. doi:10.1145/2000807.2000811.

⁷ The argument above is actually stronger: even a $(1 + \epsilon^*)$ -approximation algorithm for k -center can solve d -dimensional geometric \geq -CSP.

- 7 Rajesh Chitnis and Nitin Saurabh. Tight Lower Bounds for Approximate & Exact k -Center in \mathbb{R}^d . *CoRR*, abs/2203.08328, 2022. [arXiv:2203.08328](#).
- 8 Rajesh Hemant Chitnis, Andreas Emil Feldmann, Mohammad Taghi Hajiaghayi, and Dániel Marx. Tight Bounds for Planar Strongly Connected Steiner Subgraph with Fixed Number of Terminals (and Extensions). *SIAM J. Comput.*, 49(2):318–364, 2020.
- 9 Vincent Cohen-Addad, Arnaud de Mesmay, Eva Rotenberg, and Alan Roytman. The Bane of Low-Dimensionality Clustering. In *SODA 2018*, pages 441–456, 2018.
- 10 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 11 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, and Sudeshna Kolay. An ETH-Tight Exact Algorithm for Euclidean TSP. In *FOCS 2018*, pages 450–461, 2018. doi: 10.1109/FOCS.2018.00050.
- 12 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A Framework for Exponential-Time-Hypothesis-Tight Algorithms and Lower Bounds in Geometric Intersection Graphs. *SIAM J. Comput.*, 49(6):1291–1331, 2020. doi: 10.1137/20M1320870.
- 13 Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Fixed-parameter algorithms for (k, r) -center in planar graphs and map graphs. *ACM Trans. Algorithms*, 1(1):33–47, 2005.
- 14 Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H -minor-free graphs. *J. ACM*, 52(6):866–893, 2005.
- 15 Andreas Emil Feldmann. Fixed-parameter approximations for k -center problems in low highway dimension graphs. *Algorithmica*, 81(3):1031–1052, 2019.
- 16 Andreas Emil Feldmann and Dániel Marx. The parameterized hardness of the k -center problem in transportation networks. *Algorithmica*, 82(7):1989–2005, 2020.
- 17 Fedor V. Fomin, Sudeshna Kolay, Daniel Lokshantov, Fahad Panolan, and Saket Saurabh. Subexponential Algorithms for Rectilinear Steiner Tree and Arborescence Problems. In *SoCG 2016*, pages 39:1–39:15, 2016.
- 18 Fedor V. Fomin, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Subexponential Parameterized Algorithms for Planar and Apex-Minor-Free Graphs via Low Treewidth Pattern Covering. In *FOCS 2016*, pages 515–524, 2016.
- 19 Eli Fox-Epstein, Philip N. Klein, and Aaron Schild. Embedding Planar Graphs into Low-Treewidth Graphs with Applications to Efficient Approximation Schemes for Metric Problems. In *SODA 2019*, pages 1069–1088, 2019.
- 20 Yogesh A. Girdhar and Gregory Dudek. Efficient on-line data summarization using extremum summaries. In *ICRA 2012*, pages 3490–3496, 2012.
- 21 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985.
- 22 S. Louis Hakimi. Steiner’s problem in graphs and its implications. *Networks*, 1(2):113–133, 1971.
- 23 Christian Hennig, Marina Meila, Fionn Murtagh, and Roberto Rocci. *Handbook of cluster analysis*. CRC Press, 2015.
- 24 Dorit S. Hochbaum and David B. Shmoys. A best possible heuristic for the k -center problem. *Math. Oper. Res.*, 10(2):180–184, 1985.
- 25 Wen-Lian Hsu and George L. Nemhauser. Easy and hard bottleneck location problems. *Discret. Appl. Math.*, 1(3):209–215, 1979.
- 26 R. Z. Hwang, R. C. Chang, and Richard C. T. Lee. The Searching over Separators Strategy To Solve Some NP-Hard Problems in Subexponential Time. *Algorithmica*, 9(4):398–423, 1993.
- 27 R. Z. Hwang, Richard C. T. Lee, and R. C. Chang. The Slab Dividing Approach To Solve the Euclidean p -Center Problem. *Algorithmica*, 9(1):1–22, 1993.

- 28 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k -SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.
- 29 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
- 30 Daxin Jiang, Chun Tang, and Aidong Zhang. Cluster analysis for gene expression data: a survey. *IEEE Transactions on knowledge and data engineering*, 16(11):1370–1386, 2004.
- 31 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structural parameters, tight bounds, and approximation for (k, r) -center. *Discret. Appl. Math.*, 264:90–117, 2019.
- 32 Philip N. Klein and Dániel Marx. Solving Planar k -Terminal Cut in $O(n^{c\sqrt{k}})$ Time. In *ICALP 2012*, pages 569–580, 2012.
- 33 Philip N. Klein and Dániel Marx. A subexponential parameterized algorithm for Subset TSP on planar graphs. In *SODA 2014*, pages 1812–1830, 2014.
- 34 Daniel Lokshantov, Saket Saurabh, and Magnus Wahlström. Subexponential Parameterized Odd Cycle Transversal on Planar Graphs. In *FSTTCS 2012*, pages 424–434, 2012.
- 35 Dániel Marx. Efficient Approximation Schemes for Geometric Problems? In *ESA 2005*, pages 448–459, 2005. doi:10.1007/11561071_41.
- 36 Dániel Marx. Parameterized complexity of independence and domination on geometric graphs. In Hans L. Bodlaender and Michael A. Langston, editors, *IWPEC 2006*, pages 154–165, 2006.
- 37 Dániel Marx. A Tight Lower Bound for Planar Multiway Cut with Fixed Number of Terminals. In *ICALP 2012*, pages 677–688, 2012.
- 38 Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. On Subexponential Parameterized Algorithms for Steiner Tree and Directed Subset TSP on Planar Graphs. In *FOCS 2018*, pages 474–484, 2018.
- 39 Dániel Marx and Michal Pilipczuk. Optimal Parameterized Algorithms for Planar Facility Location Problems Using Voronoi Diagrams. In *ESA 2015*, pages 865–877, 2015.
- 40 Dániel Marx and Anastasios Sidiropoulos. The limited blessing of low dimensionality: when $1 - 1/d$ is the best possible exponent for d -dimensional geometric problems. In *SoCG 2014*, page 67, 2014.
- 41 Marie-Francine Moens, Caroline Uyttendaele, and Jos Dumortier. Abstracting of legal cases: The potential of clustering based on the selection of representative objects. *J. Am. Soc. Inf. Sci.*, 50(2):151–161, 1999.
- 42 Marcin Pilipczuk, Michal Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Subexponential-Time Parameterized Algorithm for Steiner Tree on Planar Graphs. In *STACS 2013*, pages 353–364, 2013.
- 43 Warren D. Smith and Nicholas C. Wormald. Geometric separator theorems & applications. In *FOCS 1998*, pages 232–243, 1998.
- 44 Vijay V. Vazirani. *Approximation algorithms*. Springer, 2001.

Flat Folding an Unassigned Single-Vertex Complex (Combinatorially Embedded Planar Graph with Specified Edge Lengths) Without Flat Angles

Lily Chung ✉ 

Massachusetts Institute of Technology, Cambridge, MA, USA

Erik D. Demaine ✉ 

Massachusetts Institute of Technology, Cambridge, MA, USA

Dylan Hendrickson ✉ 

Massachusetts Institute of Technology, Cambridge, MA, USA

Victor Luo ✉

Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

A foundational result in origami mathematics is Kawasaki and Justin’s simple, efficient characterization of flat foldability for unassigned single-vertex crease patterns (where each crease can fold mountain or valley) on flat material. This result was later generalized to cones of material, where the angles glued at the single vertex may not sum to 360° . Here we generalize these results to when the material forms a *complex* (instead of a manifold), and thus the angles are glued at the single vertex in the structure of an arbitrary planar graph (instead of a cycle). Like the earlier characterizations, we require all creases to fold mountain or valley, not remain unfolded flat; otherwise, the problem is known to be NP-complete (weakly for flat material and strongly for complexes). Equivalently, we efficiently characterize which combinatorially embedded planar graphs with prescribed edge lengths can fold flat, when all angles must be mountain or valley (not unfolded flat). Our algorithm runs in $O(n \log^3 n)$ time, improving on the previous best algorithm of $O(n^2 \log n)$.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Graph drawing, folding, origami, polyhedral complex, algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.29

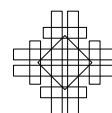
Related Version *arXiv version*: <https://arxiv.org/abs/2204.03696>

Acknowledgements We thank Joseph O’Rourke and the anonymous referees for helpful suggestions. This work grew out of an open problem session and a final project from the MIT class on Geometric Folding Algorithms: Linkages, Origami, Polyhedra (6.849) held Fall 2020.

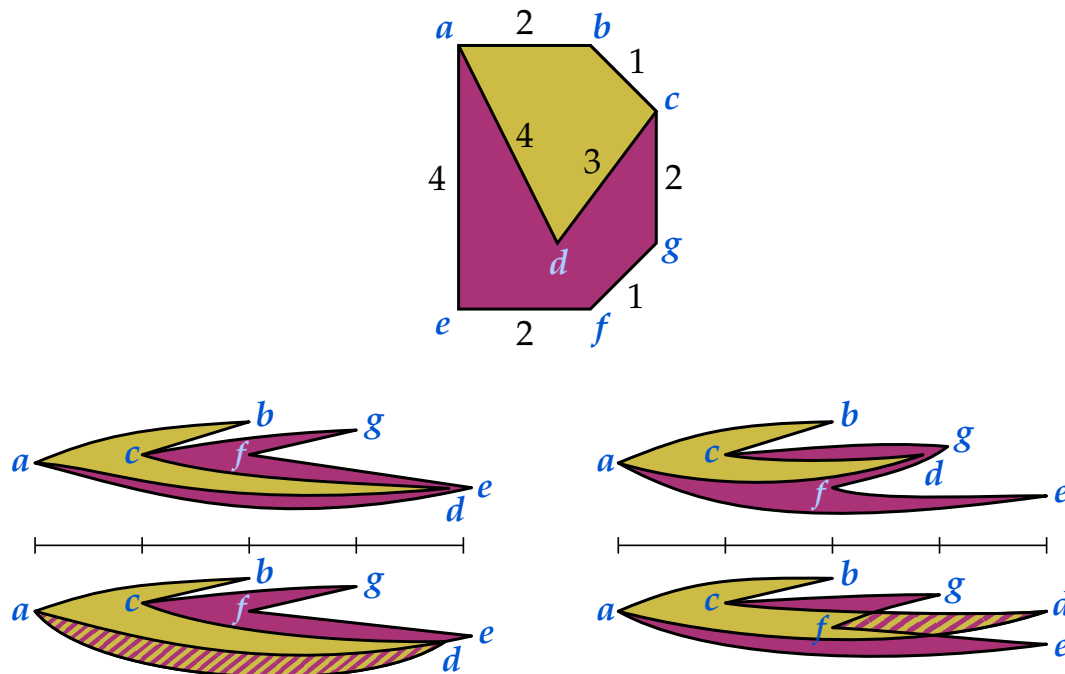
1 Introduction

The graph flat folding problem asks whether a given combinatorially embedded planar graph with prescribed edge lengths can be “folded flat” onto a line. More precisely, a *flat folding* is an assignment of x coordinates to vertices that respects the edge lengths, together with a partial order on the edges (which defines the stacking order among edges with overlapping x extents) that respects the combinatorial planar embedding and avoids crossings (edges penetrating connections between higher and lower edge endpoints, and improperly nested edge endpoint connections) [2, 9, 11].¹ Equivalently, a flat folding is a sequence or continuum

¹ In [2], flat foldings are called “linear folded states”. Here we use “flat foldings” so that they match up with the corresponding notions in computational origami.



of planar embeddings that respect the combinatorial planar embedding, avoid crossings, and converge to the correct edge lengths and to lying on a line [1, 3]. Figure 1 shows an example of a flat folding, as well as some non-examples.



■ **Figure 1** Four attempts to fold a graph with assigned edge lengths. Top left: A valid folding of the graph. Top right: Invalid because the lengths of edges ad and cd do not correspond to the original edge lengths. Bottom left: Invalid for two reasons: the cyclic ordering of edges at vertex a is not respected, and the folding exhibits incorrect layering at vertices d and e . Bottom right: Invalid since edges cross over each other.

It is known that the graph flat folding problem is strongly NP-complete in general, and solvable in linear time if all edge lengths are equal [2]. But there are two natural variations on the problem, posed in the same paper [2]. In any flat folding, we can identify the angles between consecutive edges around a vertex (as determined by the combinatorial planar embedding) as either *valley* (0°), *mountain* (360°), or *unfolded/flat* (180°). (At each vertex, these angles must sum to 360° , so there is either one mountain or two flats, and the rest are valleys.) Now we can vary two aspects of the problem:

1. What if we are also given the angle (valley/mountain/flat) between every consecutive pair of edges around each vertex?
2. What if we forbid flat angles, and instead require just valleys and mountains?

These parameters define four versions of the problem, as summarized in Table 1. The original paper [2] proved NP-completeness of the version with no angles given and allowing flat angles. Recent work shows that, if the angles are given, the problem becomes solvable in linear time (independent of whether flat angles are allowed) [3]. The remaining problem, studied here, is the version where the angles are not given, but flat angles are forbidden.

Connection to weak embeddings of graphs. Although not stated explicitly, this no-flat-angles graph flat folding problem can be solved in polynomial time by a reduction to “weak embeddings of graphs”. A key feature of this version of graph folding (in particular

■ **Table 1** Complexity of different models of graph flat folding (based on [3, Table 1], which in turn is based on open problems from [2]). Our new result is in the bottom-left.

	Flat angles forbidden	Flat angles allowed
Angles given	Linear time [3]	
Angles unspecified	$O(n^2 \log n)$ [4] \rightarrow $O(n \log^3 n)$ [new]	NP-complete [2]

distinguishing it from the NP-complete version with unknown angles that can be flat) is that the relative coordinates of the vertices are determined by the input: fixing one edge to go right from the origin, any path in the graph alternates between going right and left by the specified edge lengths, so a depth-first search fixes the vertex coordinates (and checks geometric closure constraints on cycles in the graph). The graph flat folding problem is then equivalent to asking whether this mapping from vertices to coordinates is a *weak embedding* of the graph, meaning that the vertices can be perturbed in the plane within ε -radius disks (for any $\varepsilon > 0$), and the edges can be similarly perturbed to Jordan curves within distance ε of the corresponding line segments, so that we obtain a strict embedding (no intersections except as intended at shared vertices). Recognizing weak embeddings was recently solved in $O(n^2 \log n)$ time [4],² so the same result applies to no-flat-angles graph flat folding.

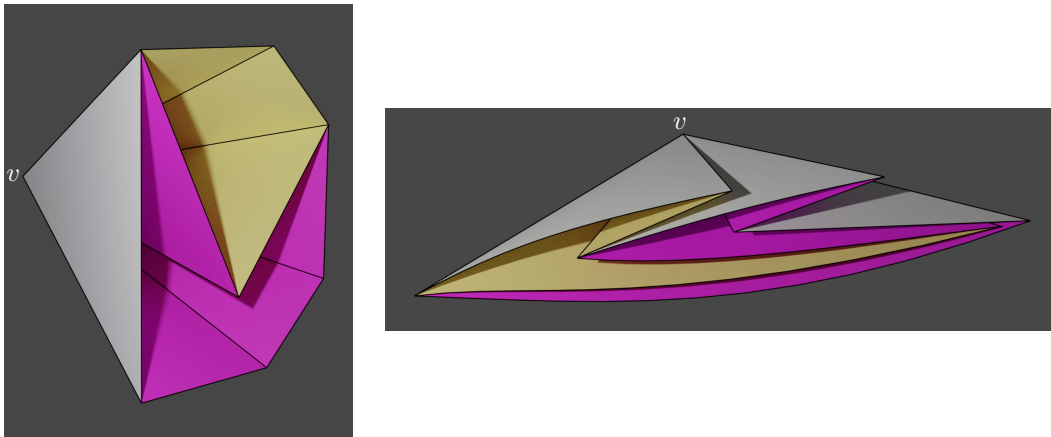
Our results. In this paper, we give a faster algorithm for the no-flat-angles graph flat folding problem. Specifically, we show how to determine whether a graph can be folded flat without flat angles in $O(n \log^3 n)$ time, which is tight up to logarithmic factors.

We extend this result to the case where some angles are specified as flat, and the problem asks to determine mountain or valley for each of the remaining angles. (The same extension also follows from the reduction to weak embedding.) Thus what makes graph flat folding hard is not the existence of flat angles, but deciding which angles are flat.

Application to single-vertex origami. The version of the graph flat folding problem we study is particularly natural when viewed from the lens of computational origami.

Define a *single-vertex complex* to consist of m polygons in 3D where the polygons all share a common vertex v , and all the shared edges between these polygons are incident to v , as in Figure 2 (left). If we intersect such a single-vertex complex with a small sphere centered at v , we obtain a planar graph embedded on the sphere, whose m edge lengths are proportional to the m polygon angles at v . In the example of Figure 2 (left), we obtain the planar graph in Figure 1 (top). A flat folding of the single-vertex complex into the plane (according to standard origami definitions [11]) corresponds to a flat folding of the combinatorially embedded planar graph with prescribed edge lengths [2, 3]. Figure 2 (right) shows such a flat folding, corresponding to the graph flat folding in Figure 1 (top left). We can similarly consider the case of a single-vertex abstract complex – that is, an abstract metric space (not embedded in 3D) formed by gluing planar polygons along edges, which all share a common vertex – together with the cyclic ordering of polygons around each shared edge. Intersecting a single-vertex abstract complex with a small intrinsic sphere centered at the shared vertex produces a graph flat folding problem, and we can construct an arbitrary combinatorially embedded planar graph with prescribed edge lengths by a suitable

² The same paper [4] develops an $O(n \log n)$ algorithm for weak embedding of graphs, but only when the given map is “simplicial”, meaning that edges do not pass through other vertices. This property does not hold in general in the graph flat folding problem.



■ **Figure 2** Unfolded and folded states of the single-vertex complex corresponding to the planar graph in Figure 1.

single-vertex abstract complex. Indeed, we can construct a multigraph in this way, so we generally allow graphs with multiple edges between the same two vertices. Therefore graph flat folding is equivalent to origami flat foldability of single-vertex (abstract) complexes.

When the planar graph is a cycle corresponding to 360° of total angle of polygons glued at a single vertex, we obtain what is known as a *single-vertex crease pattern* [11, Section 12.2]: creases emanating from a single vertex on a piece of paper. At the first OSME (Origami Science/Mathematics/Education) conference in 1989, Justin [16] and Kawasaki [17] presented characterizations of which single-vertex crease patterns fold flat: exactly those whose alternating sum of angles is zero. (A complete proof of this characterization was not published until Hull's 1994 paper [13]; see [15, Section 5.9].) Crucially, this linear-time characterization assumes that all creases must be folded either mountain or valley (none can be left unfolded flat at an angle of 180°); otherwise, single-vertex flat foldability becomes weakly NP-complete [10].

We see a similar behavior in Table 1 (bottom row), where allowing mountain, valley, and flat angles makes the problem NP-complete (even strongly), while our result shows that allowing just mountain or valley makes the problem solvable in near-linear time. Thus our result can be seen as a generalization of the Justin–Kawasaki Theorem from flat paper to complexes with similar running time. Previously, the theorem was generalized to cones of paper, where the angles sum to a value other than 360° [11, Section 12.2.1], but ours is the first generalization from manifolds to complexes with near-linear running time.

The top row of Table 1 corresponds to single-vertex *mountain-valley patterns*, where each crease is marked as mountain or valley. (Some creases could be marked unfolded/flat, but this is equivalent to removing the crease.) The previous work on given-angle complexes [3] can similarly be seen as a generalization of the previously known linear-time characterization of single-vertex mountain-valley patterns [5], [11, Section 12.2.2].

Organization. The rest of this paper is organized as follows. First we restate two needed previous results in Section 2. We then give a high-level overview of our algorithm in Section 3, and detail the various components of the algorithm in Sections 4, 5, 6, and 7.

2 Background

Our results rely on two previous results, which we restate here for completeness.

First, based on results of Hull [14], Demaine and O'Rourke [11] characterized the flat-foldable mountain/valley assignments of a cycle, which we will apply to each face in a connected combinatorially embedded planar graph:

► **Lemma 1** ([11, Corollary 12.2.12]). *Let f be a simple cycle with edge lengths $\theta_1, \dots, \theta_n$. If the edge lengths are all equal, then a crease assignment on f is flat foldable in precisely the following cases:*

- Case A: *The cycle f is an interior face with equal-length edges, and there are exactly 2 more valley folds than mountain folds.*
- Case B: *The cycle f is an exterior face with equal-length edges, and there are exactly 2 more mountain folds than valley folds.*

Otherwise, take any maximal sequence e_m, \dots, e_{m+k-1} of k contiguous equal-length edges surrounded by strictly longer edges, so that³

$$\theta_{m-1} > \theta_m = \dots = \theta_{m+k-1} < \theta_{m+k}$$

Then a crease assignment is flat foldable in precisely the following cases:

- Case C: *k is odd, and there are an equal number of mountain and valley folds incident to edges e_m, \dots, e_{m+k-1} . Additionally, replacing all of the edges e_{m-1}, \dots, e_{m+k} with a single edge of length $\theta_{m-1} - \theta_m + \theta_{m+k}$ yields a flat-foldable face with the same crease assignment.*
- Case D: *k is even, and the numbers of mountain and valley folds incident to edges e_m, \dots, e_{m+k-1} differ by ± 1 . Additionally, replacing all of the edges e_m, \dots, e_{m+k-1} with a single new vertex yields a flat-foldable face, where the crease assignment is the same except that it assigns the new vertex to be the same type as the majority of the folds incident to e_m, \dots, e_{m+k-1} . (That is, the new vertex is a mountain fold in this assignment if the number of mountain folds was 1 greater than the number of valley folds.)*

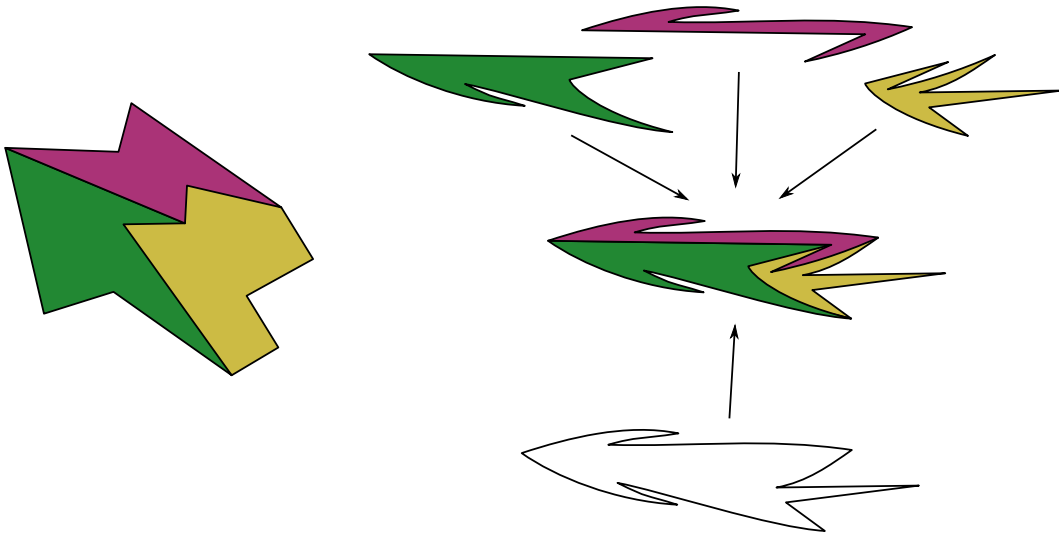
Second, Abel et al. [3] proved that a flat folding of a graph is equivalent to a compatible folding of each face:⁴

► **Theorem 2** ([3, Theorem 2]). *Let G be a connected multigraph with an assignment of measures to every angle in G . That is, for each angle a we are given its measure $m_a \in \{0^\circ, 180^\circ, 360^\circ\}$. Suppose that, for every face f , the restriction of this assignment to f yields a flat-foldable mountain-valley assignment when f is treated as a simple cycle. Suppose also that the assignment is **compatible** in that the sum of angles around each vertex is equal to 360° . Then there exists a flat folding of G whose angles have the assigned measures.*

Figure 3 shows an example of combining compatible flat foldings of individual faces to obtain a flat folding of the entire graph.

³ The indices should be understood as being modulo n .

⁴ A similar style of result (“faces being valid implies global validity”) was obtained in the context of upward drawings of graphs [6, Theorem 3]. It also does not allow flat angles. That result, however, does not deal with prescribed edge lengths, which significantly complicates whether faces are flat foldable.



■ **Figure 3** Left: A graph with three interior faces. Right: Given compatible flat foldings of all four faces, a flat folding of the graph can be generated.

3 Algorithm Overview

In this section, we provide a high-level outline of our algorithm for determining whether a connected combinatorially embedded planar (multi)graph with prescribed edge lengths can be folded flat. This algorithm takes as input a combinatorial embedding of the graph G (which we allow to have multiple edges between the same two vertices) and an assignment of lengths to the edges of G . We assume for now that the graph is connected and that we are given a single face of G designated as the *exterior*; in Section 7, we will remove both constraints.

By Theorem 2, determining whether such a graph has a flat folding is equivalent to determining whether there are compatible flat-foldable crease assignments for each face. To accomplish this, we reduce the graph flat folding problem to a boolean constraint satisfaction problem, with constraints deriving both from the requirement that each face needs to be flat foldable and from the compatibility requirement between faces. The variables will correspond to angles in each face of the graph (including some angles only present in virtual intermediate states), and indicate whether that angle is a valley fold or a mountain fold. The resulting constraint satisfaction problem has the following structure:

- Each clause specifies an exact number of true variables in some set.
- Each variable appears in exactly two clauses.
- The graph whose vertices are clauses and whose edges are variables, connecting the two clauses in which each variable appears, is bipartite. In particular, each vertex appears in exactly one clause on each side of the bipartition.
- The same graph is planar.

We call such a problem *planar bipartite positive *-in-*SAT-E2*. This terminology generalizes the standard notion of “positive i -in- k SAT” [12, 18] where every clause requires satisfying exactly i out of (up to) k variables, which are never negated (hence “positive”), to the situation where number of variables and required true variables may vary from clause to clause. The standard suffix “-E2” represents the requirement that every variable appears in exactly two clauses [12, 8]. The “planar” prefix is also standard [12, 18], while the “bipartite” prefix is new (and makes sense only with the “-E2” requirement).

In Section 4, we describe the constraints which express that each face must be folded flat. In Section 5, we describe the constraints capturing compatibility between faces, and prove that the resulting constraint problem is equivalent to flat folding the graph. In Section 6, we show that planar bipartite positive **-IN-*SAT-E2* can be solved in $O(n \log^3 n)$ time through a reduction to a flow problem. In Section 7, we put the pieces together to obtain our main result, and describe three extensions: to graphs with some prescribed flat angles, to disconnected graphs, and to graphs with unknown exterior face.

4 Single Face Constraints

In this section, we describe the constraints obtained from the requirement that each face of the graph is folded flat. Although faces of the graph may not be bound by simple cycles (in the case of cut vertices), there exists a simple cycle corresponding to each face. This cycle can be constructed by enumerating the face's incident edges and angles in order, duplicating any repeated vertices or edges. Although there may exist flat foldings of this simple cycle which do not correspond to flat foldings of the original face, Theorem 2 tells us that compatible flat foldings of the corresponding simple cycles are sufficient for flat foldability of the full graph. From here on, when we discuss flat foldability of an individual face, we will actually be referring to flat foldability of the corresponding simple cycle.

Now consider flat folding a single face f . We check (and henceforth assume) that the edge lengths satisfy the basic closure property (mentioned in Section 1) that the number of edges is even and the alternating sum of edge lengths is zero; otherwise, flat folding is impossible. It remains to determine flat-foldable mountain/valley assignments.

We introduce a boolean variable x_a for each angle a in f . These variables represent an assignment of creases: $x_a = 0$ if a is a valley fold (0°) and $x_a = 1$ if a is a mountain fold (360°). We present an algorithm which, given the edge lengths and interior/exterior assignment of f and a variable x_a assigned to each angle a in f , generates a set of constraints C_f on the variables x_a , possibly introducing additional variables, such that solutions to this constraint problem correspond to flat-foldable crease assignments of f . The constraints are of the form “exactly c variables from a set S are true,” which we write

$$\sum_{x \in S} x = c.$$

Additionally, each constraint generated will be colored either **red** or **blue**; this coloring will be used later to show that the constraint satisfaction problem is bipartite. The algorithm essentially follows Lemma 1:

- If all edges of f have equal length, then let V be the set of all angles in f , and let b be -1 if f is an interior face, or $+1$ if f is an exterior face. Generate just the **red** constraint

$$\sum_{a \in V} x_a = \frac{|V|}{2} + b. \quad (1)$$

- Otherwise, not all of the edges of f have equal length. Find a sequence e_m, \dots, e_{m+k-1} of k consecutive equal-length edges, such that $\theta_{m-1} > \theta_m = \dots = \theta_{m+k-1} < \theta_{m+k}$; this is guaranteed to exist by considering a maximal sequence of consecutive edges with minimum length. Let S be the set of angles in f incident to e_m, \dots, e_{m+k-1} .
- If k is odd (i.e., $|S|$ is even), generate the **red** constraint

$$\sum_{a \in S} x_a = \frac{|S|}{2}. \quad (2)$$

29:8 Flat Folding an Unassigned Single-Vortex Complex Without Flat Angles

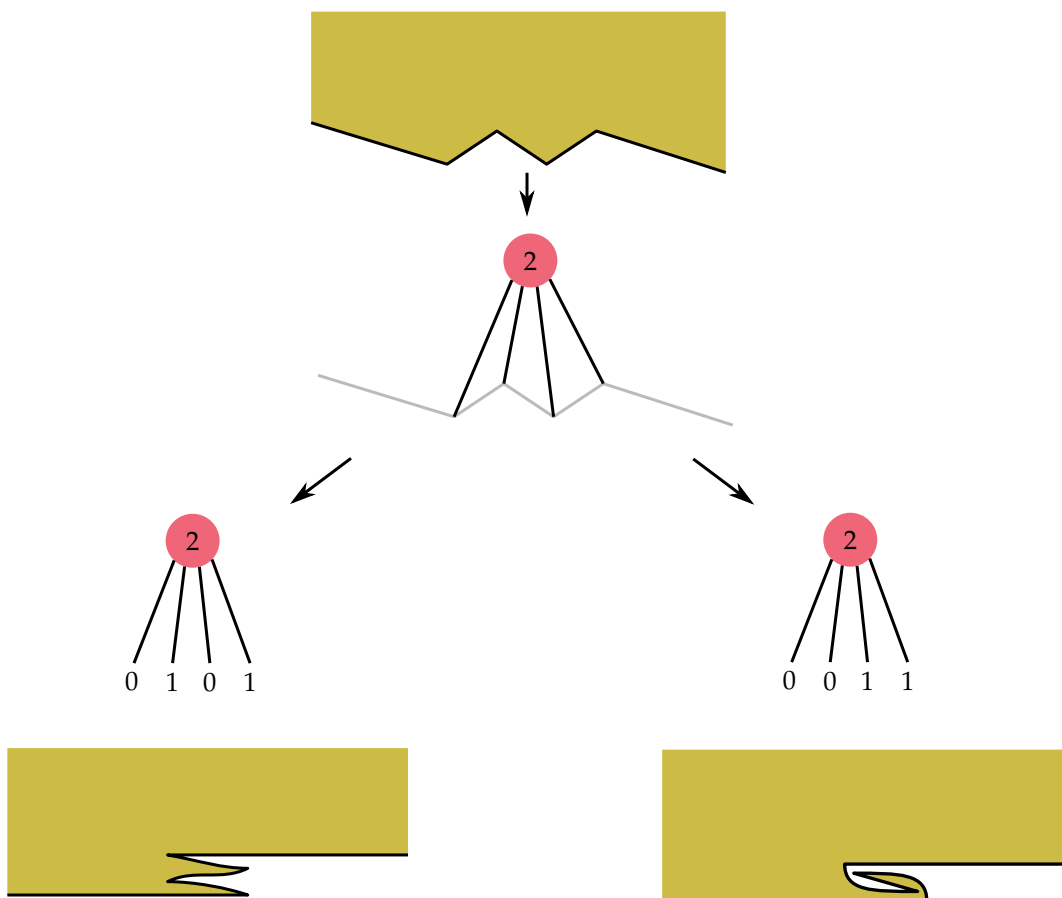
Having done so, replace all the edges e_{m-1}, \dots, e_{m+k} with a single edge of length $\theta_{m-1} - \theta_m + \theta_{m+k}$ to construct a smaller face f' , and recursively output the constraints in $C_{f'}$.

- If instead k is even (i.e., $|S|$ is odd), introduce two fresh boolean variables y and z , and generate the following **red** and **blue** (respectively) constraints:

$$y + \sum_{a \in S} x_a = \frac{|S|+1}{2}; \quad (3)$$

$$y + z = 1. \quad (4)$$

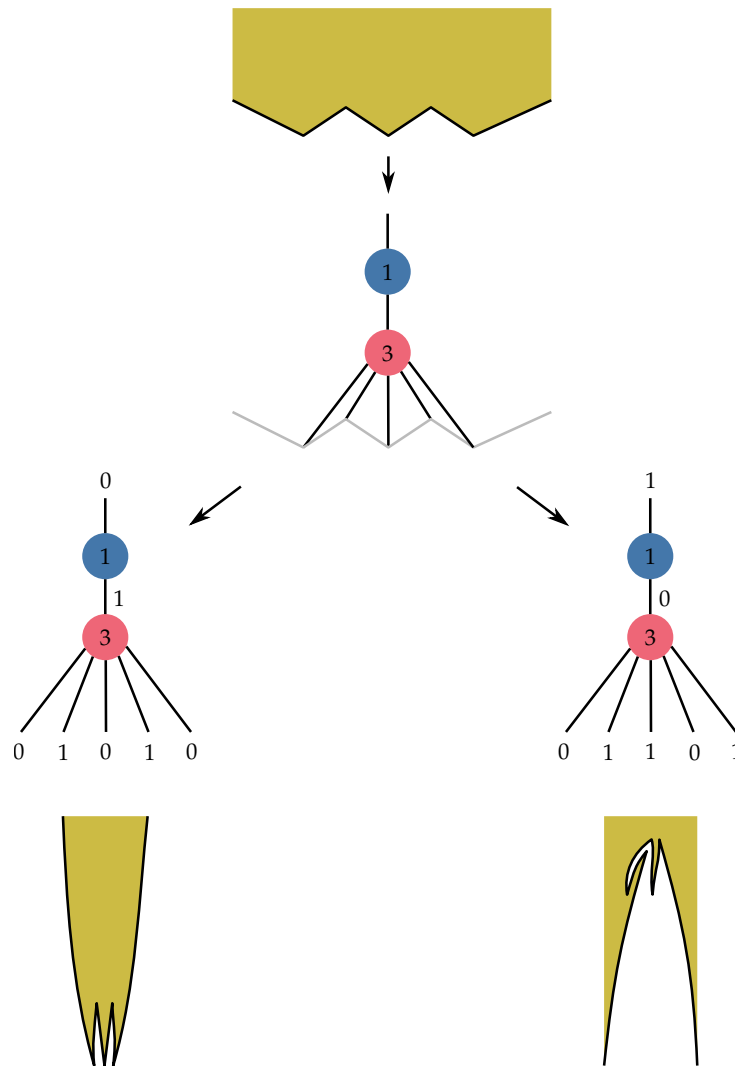
Then replace all the edges e_m, \dots, e_{m+k-1} with a single new angle whose associated variable is z to construct a smaller face f' , and recursively output the constraints in $C_{f'}$.



■ **Figure 4** The case where k is odd. In this diagram, circular vertices represent constraints and edges represent boolean variables. Top: The **red** constraint expresses that exactly half of the creases must be mountain folds and the others must be valley folds. Bottom: Two possible satisfying variable assignments and the associated local foldings.

We now show that solutions to the constraints generated by this algorithm correspond to flat-foldable crease assignments of f .

► **Theorem 3.** *A mountain/valley assignment for f is flat foldable if and only if it can be extended to a satisfying assignment of C_f .*



■ **Figure 5** The case where k is even. The pair of constraints expresses that the number of mountain folds and valley folds must differ by 1, and the majority value is equal to the newly generated variable.

We may need to extend the assignment to account for variables introduced in the case where k is even. The values for these variables are forced by the constraints added when the variables are introduced, and can be determined by considering variables in the order they were added.

Proof. The proof is by induction on the size of f . If all the edges of f have equal length, then it is immediate by cases A and B of Lemma 1 that an assignment is flat foldable if and only if equation (1) holds.

When the edges are not all equal in length, the algorithm finds some maximal sequence e_m, \dots, e_{m+k-1} of k equal-length edges surrounded by strictly longer edges, whose incident angles we call S .

If k is odd, then by case C of Lemma 1, the assignment is flat foldable for f if and only if it both assigns an equal number of mountain and valley folds to the angles in S , and is also a flat-foldable crease assignment for f' , where f' is the face resulting from replacing the

edges e_{m-1}, \dots, e_{m+k} with a single edge of length $\theta_{m-1} - \theta_m + \theta_{m+k}$. The first condition is just equation (2), and the second is equivalent by the inductive hypothesis to the set of constraints $C_{f'}$ obtained by recursion on f' . An example of this case is shown in Figure 4.

If k is even, then by case D of Lemma 1, the assignment is flat foldable if and only if the mountain and valley folds assigned to the angles in S differ by 1, and it is also a flat-foldable crease assignment for f' when suitably extended. Here f' is the face resulting from replacing the edges e_m, \dots, e_{m+k-1} with a single angle, and the assignment is extended to assign the new angle to be the same type as the majority of the folds it assigned to the angles in S . Equation (3) constrains the number of folds to differ by 1, where y is the minority fold type, and equation (4) constrains z to be the opposite of y , so z is the majority fold type. By the inductive hypothesis, the constraints $C_{f'}$ obtained by recursion on f' are equivalent to the statement that f' is flat foldable under the assignment extended to assign z to the new angle. An example of this case is shown in Figure 5.

In all cases, we find that the assignment is flat foldable if and only if it satisfies the constraints. ◀

We also show that these constraints can be computed efficiently and satisfy certain properties which will be useful for solving them.

► **Theorem 4.** *The algorithm for computing C_f takes time linear in the number of angles in f . The variables and clauses of C_f form a graph in which graph vertices correspond to clauses and graph edges correspond to variables, when an additional blue graph vertex is added for each angle of f . Then this graph is a bipartite (i.e. 2-colored) forest with linearly many vertices, and there is a planar embedding of this graph within f such that each vertex corresponding to an angle of f is located at the vertex of f incident to that angle.*

Proof. Let n be the number of angles in f . We prove by induction that C_f forms a graph as described with at most $2n$ vertices.

In the case where the edges all have equal length, C_f is a star graph whose central vertex is a red clause and whose outer vertices are the blue angles of f , so it is a bipartite forest with $n + 1 \leq 2n$ vertices. The planar embedding can be achieved by placing the central vertex within f and drawing edges to all the vertices of f .

When the edges of f are not all equal in length, the algorithm finds some sequence of k edges whose $k + 1$ incident angles we call S . Let T be the star graph whose central vertex is the red clause added in this step and whose outer vertices are the blue angles of S ; this is a bipartite forest with $k + 2$ vertices.

When k is odd, the graph C_f is simply the disjoint union of $C_{f'}$ and T , where f' is a face with $n - k - 1$ angles. By the inductive hypothesis $C_{f'}$ is a bipartite forest with at most $2(n - k - 1)$ vertices, so C_f is a bipartite forest with at most $k + 2 + 2(n - k - 1) = 2n - k \leq 2n$ vertices. The planar embedding of C_f is obtained from the planar embedding of $C_{f'}$ by simply placing T alongside it; none of the edges need to cross because the angles in S are contiguous in f .

When k is even, the graph C_f is formed from the disjoint union of $C_{f'}$ and T by adding an edge from the red central vertex of T to the blue vertex corresponding to some angle a' of f' , where f' is a face with $n - k$ angles. By the inductive hypothesis $C_{f'}$ is a bipartite forest with at most $2(n - k)$ vertices, so C_f is a bipartite forest with at most $k + 2 + 2(n - k) = 2n - k + 2 \leq 2n$ vertices. The planar embedding of C_f is obtained from the planar embedding of $C_{f'}$ by first placing T alongside it as before; again none of the edges cross because S is contiguous in f . Then the edge from the central vertex of T to the vertex corresponding to a' can be added without crossing because a' occurs in the same place in f' 's cyclic order of angles as S does in f .

Thus C_f is a linear-sized bipartite forest with the desired planar embedding. We need to show that it can be computed in linear time. It is straightforward to charge the work performed by the algorithm at each step to the newly created vertices, except for finding the sequence e_m, \dots, e_{m+k-1} of equal-length edges surrounded by strictly longer edges. We cannot accomplish this by simply scanning through the edges of the face at each iteration, since this would take linear time and there might be linearly many iterations. We instead solve this by maintaining a cyclic doubly-linked list C , each of whose entries corresponds to a maximal contiguous sequence of equal-length edges. Additionally we keep a list M of such entries of C which are surrounded by longer entries. These can be computed once at the beginning of the algorithm in linear time, and then maintained at each iteration. At each iteration a sequence e_m, \dots, e_{m+k-1} is obtained by taking the first entry from M and removing it from both M and C . When the new face f' is computed, we add any new edges to C and check whether any of the newly adjacent pairs of entries have equal length; if so we consolidate them into a single entry of C . We also check whether any of the newly adjacent entries have become surrounded by strictly longer entries; if so we add them to M . These checks take constant time in each iteration since at most two new pairs of adjacent entries can be created. So computing C_f takes linear time overall. ◀

5 Compatibility Constraints

Next, we describe the constraints needed to ensure that the crease assignments are compatible between faces. The angles around each vertex must sum to 360° ; this means exactly one of these angles is a mountain fold, as shown in Figure 6. So for each vertex v of the graph, we generate a blue constraint C_v :

$$\sum_{a \in A_v} x_a = 1, \quad (5)$$

where A_v is the set of angles incident to v .

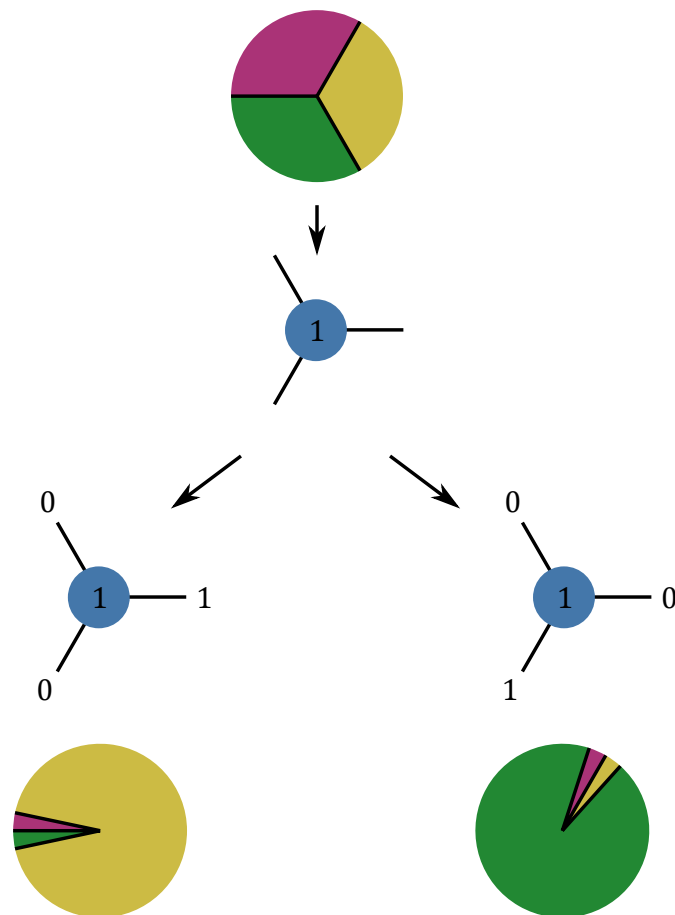
► **Theorem 5.** *A connected combinatorially embedded planar multigraph with prescribed edge lengths has a flat folding with no flat angles if and only if the constraint satisfaction problem consisting of*

- for each face f , the constraints C_f described in Section 4, and
- for each vertex v , the constraint C_v described above

is satisfiable. Moreover, these constraints can be computed in time linear in the number of angles in the graph.

Proof. Suppose the graph has such a flat folding, and assign variables representing angles in the graph based on whether the corresponding angle is a mountain or a valley fold in the flat-folded state; this is only a partial assignment since some variables do not correspond to angles of the original graph. Each face (and thus its corresponding simple cycle) is folded flat, and the variables which are not yet assigned are disjoint between faces, so by Theorem 3 we can extend the assignment to an assignment of all variables which satisfies C_f for every face f . The assignment also satisfies C_v since exactly one angle incident to v has measure 360° in the folded state.

Conversely, suppose there is a satisfying assignment. Then assign each angle to be mountain or valley based on the value of the corresponding variable. By Theorem 3, this gives a flat-foldable crease assignment for each face. These crease assignments are compatible because the variable assignments satisfy each C_v , so by Theorem 2 there is a flat folding with these angle assignments.



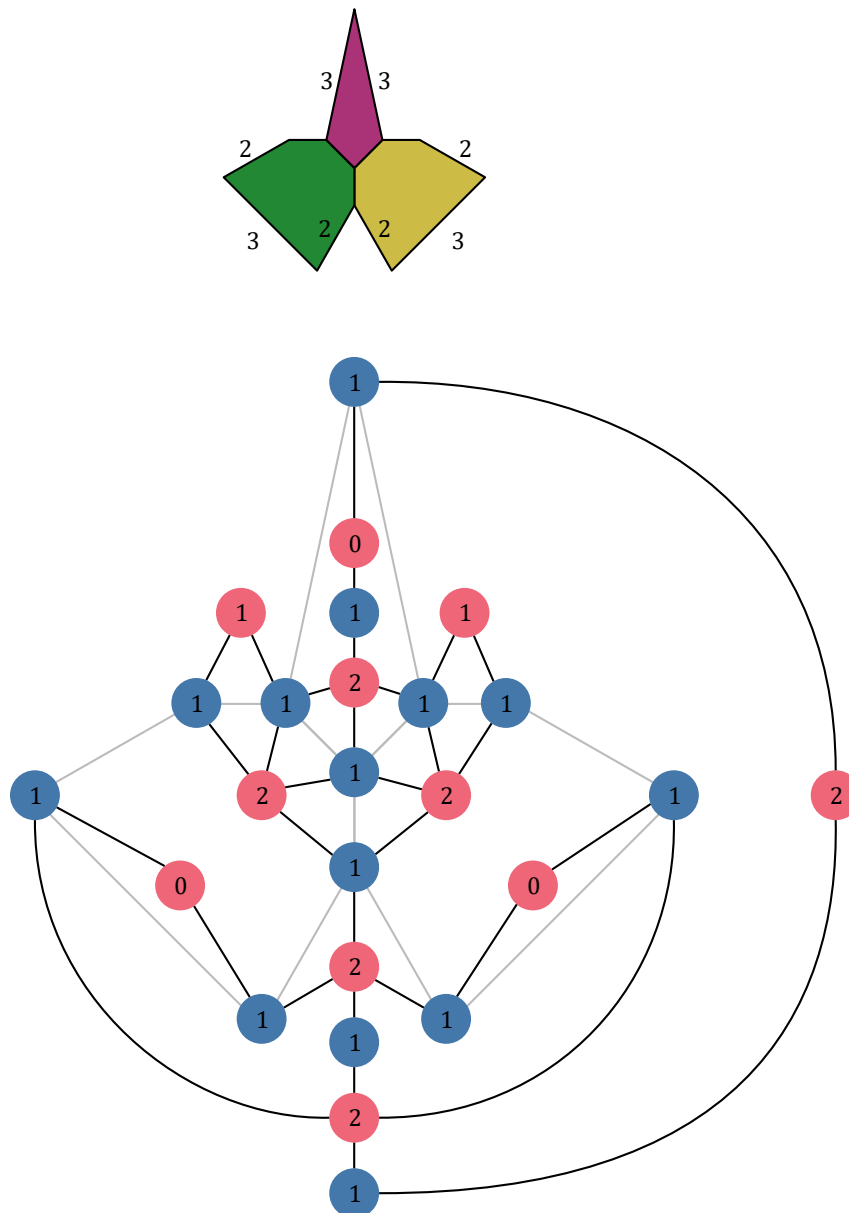
■ **Figure 6** A vertex folds flat under a given crease assignment if and only if exactly one of the incident angles is a mountain fold.

Finally, we show that the set of constraints can be computed in linear time. By Theorem 4 each set of face constraints C_f can be computed in time linear in the number of angles incident to f . Since the sets of angles incident to different faces are disjoint, it takes linear time overall to compute the face constraints. Similarly, computing each vertex constraint C_v takes time linear in the number of angles incident to v , and these are all disjoint from each other as well. So the set of constraints can be computed in time linear in the number of angles of the graph. ◀

6 Solving the Constraint Satisfaction Problem

What remains is solving the constraint satisfaction problem consisting of C_f and C_v for each face and vertex of the graph. Inspecting the constraints reveals that they are an instance of planar bipartite positive $\ast\text{-IN-}\ast\text{SAT-E2}$:

- Each constraint has the form $\sum_{x \in S} x = c$ for some set S of variables and constant c ; this is a clause saying exactly c variables in S are true.
- The red and blue clauses provide the bipartition. Each variable is in exactly one red clause and exactly one blue clause. For each angle a incident to a face f and a vertex v , the variable x_a appears in one red clause belonging to C_f and one blue clause C_v . All other variables satisfy this condition because the subgraph corresponding to each C_f is bipartite according to Theorem 4.



■ **Figure 7** Top: An example graph with assigned edge lengths. Unlabeled edges have length 1. Bottom: The resulting instance of planar bipartite positive $\ast\text{-IN-}\ast\text{SAT-E2}$ (overlaid on the original graph in gray). Since this instance is unsatisfiable, the original graph cannot be folded flat.

- The graph corresponding to the constraint satisfaction problem is planar. We can place each clause C_v at the corresponding vertex v . Then for each face f we can place the graph corresponding to C_f inside f ; by Theorem 4 this can be done without violating planarity. An example of the planar embedding constructed for the entire constraint satisfaction problem is shown in Figure 7.

All that remains to be shown is that planar bipartite positive $\ast\text{-IN-}\ast\text{SAT-E2}$ can be solved efficiently. We now describe a fairly standard reduction to a max-flow problem, which can be solved in near-linear time.

► **Theorem 6.** *Planar bipartite positive *-IN-*SAT-E2 can be solved in $O(n \log^3 n)$ time, where n is the number of clauses.*

Proof. We use the graph with clauses as vertices and variables as edges, as described earlier and shown in Figure 7. For each red clause r which expects ℓ_r true variables, we add a new source vertex and an edge from the source vertex to r with capacity ℓ_r . Similarly, for every blue clause b expecting ℓ_b true variables, we add a new sink vertex and an edge from b to the sink vertex with capacity ℓ_b . Finally, we assign a capacity of 1 to each edge corresponding to a variable, which goes from a red clause to a blue clause. This gives us an instance of multi-source multi-sink planar max-flow, for which the maximum possible flow can be determined in time $O(k \log^3 k)$ [7] where k is the number of vertices in the flow graph. Since the flow graph has exactly twice as many vertices as there were clauses, the maximum flow can be determined in time $O(n \log^3 n)$.

We will assume that

$$T := \sum_{\text{red } r} \ell_r = \sum_{\text{blue } b} \ell_b,$$

since this is clearly required for the constraint problem to be satisfiable.

To solve the constraint satisfaction problem, we ask if the maximum flow has value T ; this is clearly an upper bound on the maximum flow.

An integer flow uses some set of edges corresponding to variables, which specifies an assignment. The flow constraint on the edges to the appropriate source or sink forces the flow to use at most ℓ_c variables in clause c , and in order to reach the target flow T we must use exactly this many variables in each clause. Thus the desired flow exists if and only if the instance of planar bipartite positive *-IN-*SAT-E2 is solvable. ◀

7 Putting Things Together

Combining Theorem 5 and Theorem 6 immediately gives our main result:

► **Corollary 7.** *We can determine whether a connected combinatorially embedded planar multigraph with prescribed edge lengths and exterior face has a flat folding with no flat angles in $O(n \log^3 n)$ time, where n is the number of angles in the graph.*

Proof. The constraint problem instance can be computed in linear time, and so it has linearly many clauses, which can thus be solved in time $O(n \log^3 n)$. ◀

This result can be extended in three ways, described next.

7.1 Extension to Specified Flat Angles

First, we can allow flat (180°) angles in the folded graph, provided the input specifies *which* angles are flat, leaving the remaining angles free to be mountain or valley (but not flat).

To accomplish this, first observe that for there to be a flat folding, each vertex must have exactly zero or two flat angles. We do not create variables for flat angles, since their angle is already known. Within a face that contains a flat angle, we treat the two edges around the flat angle as a single longer edge. At a vertex v which has two flat angles, we need all other angles to be valley, so the constraint C_v is now

$$\sum_{a \in A_v} x_a = 0, \tag{6}$$

where A_v includes only non-flat angles at v . The rest of the algorithm is as before.

7.2 Extension to Disconnected Graphs

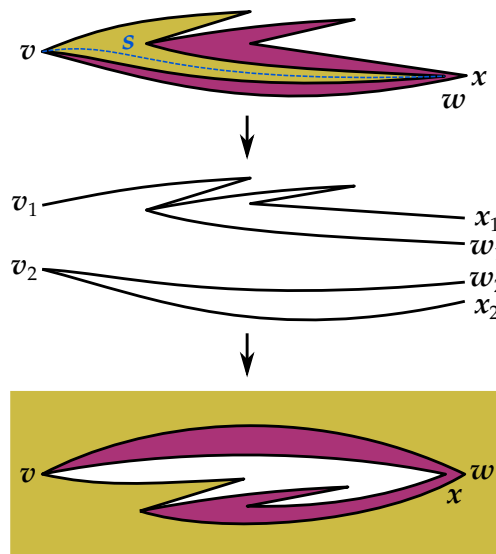
Second, we can account for the case where the graph is disconnected. Here we assume that the connected components are arranged in a rooted forest (i.e., a collection of rooted trees), where each non-root component specifies which interior face of its parent it is to reside in. This condition can arise from folding an arbitrary single-vertex complex, where some faces share the central vertex but no edges; then the structure of the complex requires a certain arrangement of components within faces. We first check that each connected component is foldable. If this is the case, then the only obstacle to foldability is being able to fit the folded state of each child graph G_i inside the designated face p_i of its parent.

We define the *folded diameter* of a graph or face to be the maximum distance between any pair of vertices in its folded state. Since the locations of all flat angles are specified, the relative vertex coordinates are determined and can be computed in linear time (as described in Section 1), so this value can be computed easily without knowledge of the folding. It turns out that we can fit G_i inside p_i in a folding if and only if the folded diameter of G_i is at most the folded diameter of p_i . To show this, we can imagine applying cases C and D of Lemma 1 to p_i repeatedly until all the edge lengths are equal. Because the face transformations in those cases preserve folded diameter, it follows that the remaining edges all have length equal to the folded diameter of p_i . Thus, if the folded diameter of G_i is less than or equal to the length of one of these edges, we can place a folding of G_i along it in the folded state. On the other hand, if the folded diameter of G_i is greater than the folded diameter of p_i , then we can clearly never fit a folding of G_i inside a folding of p_i .

7.3 Finding an Exterior Face

Third, instead of assuming that the exterior face is given, we can determine in linear time a face that is a suitable exterior face if any face is. Observe that the exterior face must be *full-diameter* in the sense that its folded diameter (defined in Section 7.2 above) equals the folded diameter of the entire graph, because some vertex of the minimum (and maximum) coordinate must be on the exterior face in any flat folding. We claim that *every* full-diameter face is an equally suitable exterior face: if there is a flat folding with any one full-diameter face as exterior face, then there is a flat folding with any desired full-diameter face as exterior face. Thus, to determine whether the graph is flat foldable, we can simply find any full-diameter face and specify it as the exterior face.

To prove the claim, consider a flat folding of the graph, say with exterior face e . Take any non-exterior full-diameter face f , with diameter realized by vertices v and w . Face f consists of two folded paths connecting v and w . By the argument in Section 7.2 above, in any folding of f resulting from Lemma 1, we can select v and w such that the two folded paths are separable: we can draw a straight line segment s from v to w that is layered in between the two folded paths. Because s is full diameter, it partitions the edges of the graph into two halves H_1 and H_2 , where H_1 is entirely before H_2 in the layer order. Some vertices (including v and w) have some incident edges in H_1 and other incident edges in H_2 . We can imagine splitting each such vertex x into two vertices x_1 and x_2 , where x_i is incident to the edges that lie within H_i , so that H_1 and H_2 become disconnected from each other. We then swap the layer order of the two halves, placing H_2 before H_1 , and for each split vertex x , reconnect the two halves x_2 and x_1 , which corresponds to a cyclic shift of the edges incident to x . This process is illustrated in Figure 8. Intuitively, we can view the folding as lying on an American football (prolate spheroid), where the two poles represent the minimum and maximum vertex coordinates; then this transformation corresponds to spinning the line along which we cut this football open to define the extremes in the other dimension (layer order). Thus we still obtain a flat folding of the graph, but now f is the exterior face.



■ **Figure 8** Cutting a flat folding apart and reassembling it with a different exterior face.

7.4 Finale

Putting these extensions together, we have the following more general result:

► **Corollary 8.** *Given a combinatorially embedded planar multigraph with prescribed edge lengths and some angles specified as flat, we can determine in $O(n \log^3 n)$ time whether there is a flat folding that has precisely the specified angles flat.*

On the other hand, if the set of flat angles is not specified, it is NP-complete to determine whether there is a flat folding [2], so this implies that the hard part is deciding which angles should be flat.

References


- 1 Timothy G. Abbott, Erik D. Demaine, and Blaise Gassend. A generalized carpenter’s rule theorem for self-touching linkages. *arXiv:0901.1322*, 2009. [arXiv:0901.1322](#).
- 2 Zachary Abel, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Jayson Lynch, Tao B. Schardl, and Isaac Shapiro-Elowitz. Folding equilateral plane graphs. *International Journal of Computational Geometry and Applications*, 23(2):75–92, April 2013.
- 3 Zachary Abel, Erik D. Demaine, Martin L. Demaine, David Eppstein, Anna Lubiw, and Ryuhei Uehara. Flat foldings of plane graphs with prescribed angles and edge lengths. *Journal of Computational Geometry*, 9(1):74–93, 2018.
- 4 Hugo A. Akitaya, Radoslav Fulek, and Csaba D. Tóth. Recognizing weak embeddings of graphs. *ACM Transactions on Algorithms*, 15(4), October 2019. [doi:10.1145/3344549](#).
- 5 Marshall Bern and Barry Hayes. The complexity of flat origami. In *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 175–183, Atlanta, January 1996.
- 6 Paola Bertolazzi, Giuseppe Di Battista, Giuseppe Liotta, and Carlo Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(6):476–497, 1994.
- 7 Glencora Borradaile, Philip N. Klein, Shay Mozes, Yahav Nussbaum, and Christian Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. *SIAM Journal on Computing*, 46(4):1280–1303, 2017. [doi:10.1137/15M1042929](#).

- 8 M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation*, 206(11):1264–1275, 2008. doi: 10.1016/j.ic.2008.07.003.
- 9 Robert Connelly, Erik D. Demaine, and Günter Rote. Infinitesimally locked self-touching linkages with applications to locked trees. In J. Calvo, K. Millett, and E. Rawdon, editors, *Physical Knots: Knotting, Linking, and Folding of Geometric Objects in 3-space*, pages 287–311. American Mathematical Society, 2002.
- 10 Erik D. Demaine. 6.849: Geometric folding algorithms: Linkages, origami, polyhedra: Lecture 5. MIT class, fall 2010. URL: <https://courses.csail.mit.edu/6.849/fall10/lectures/L05.html?notes=4>.
- 11 Erik D. Demaine and Joseph O’Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, 2007.
- 12 Ivan Tadeu Ferreira Antunes Filho. Characterizing boolean satisfiability variants. M.eng. thesis, Massachusetts Institute of Technology, 2019. URL: <https://erikdemaine.org/theses/ifilho.pdf>.
- 13 Thomas Hull. On the mathematics of flat origamis. *Congressus Numerantium*, 100:215–224, 1994.
- 14 Thomas Hull. The combinatorics of flat folds: a survey. In *Origami³: Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, pages 29–38, Monterey, California, March 2001.
- 15 Thomas C. Hull. *Origametry: Mathematical Methods in Paper Folding*. Cambridge University Press, December 2020.
- 16 Jacques Justin. Aspects mathématiques du pliage de papier (Mathematical aspects of paper folding). In H. Huzita, editor, *Proceedings of the 1st International Meeting of Origami Science and Technology*, pages 263–277, Ferrara, Italy, December 1989. Originally appeared in *L’Owert*, number 47, 1987, pages 1–14. URL: <https://publimath.univ-irem.fr/biblio/IST87008.htm>.
- 17 Toshikazu Kawasaki. On the relation between mountain-creases and valley-creases of a flat origami. In H. Huzita, editor, *Proceedings of the 1st International Meeting of Origami Science and Technology*, pages 229–237, Ferrara, Italy, December 1989. An unabridged Japanese version appeared in *Sasebo College of Technology Report*, 27:153–157, 1990.
- 18 Wolfgang Mulzer and Günter Rote. Minimum-weight triangulation is NP-hard. *Journal of the ACM*, 55(2), May 2008. doi:10.1145/1346330.1346336.

Hop-Spanners for Geometric Intersection Graphs

Jonathan B. Conroy ✉

Department of Computer Science, Tufts University, Medford, MA, USA

Csaba D. Tóth ✉ 

Department of Mathematics, California State University Northridge, Los Angeles, CA, USA

Department of Computer Science, Tufts University, Medford, MA, USA

Abstract

A t -spanner of a graph $G = (V, E)$ is a subgraph $H = (V, E')$ that contains a uv -path of length at most t for every $uv \in E$. It is known that every n -vertex graph admits a $(2k - 1)$ -spanner with $O(n^{1+1/k})$ edges for $k \geq 1$. This bound is the best possible for $1 \leq k \leq 9$ and is conjectured to be optimal due to Erdős' girth conjecture.

We study t -spanners for $t \in \{2, 3\}$ for geometric intersection graphs in the plane. These spanners are also known as *t-hop spanners* to emphasize the use of graph-theoretic distances (as opposed to Euclidean distances between the geometric objects or their centers). We obtain the following results: (1) Every n -vertex unit disk graph (UDG) admits a 2-hop spanner with $O(n)$ edges; improving upon the previous bound of $O(n \log n)$. (2) The intersection graph of n axis-aligned fat rectangles admits a 2-hop spanner with $O(n \log n)$ edges, and this bound is the best possible. (3) The intersection graph of n fat convex bodies in the plane admits a 3-hop spanner with $O(n \log n)$ edges. (4) The intersection graph of n axis-aligned rectangles admits a 3-hop spanner with $O(n \log^2 n)$ edges.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics; Mathematics of computing → Paths and connectivity problems; Theory of computation → Computational geometry

Keywords and phrases geometric intersection graph, unit disk graph, hop-spanner

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.30

Related Version *Full Version*: <https://arxiv.org/abs/2112.07158>

Funding *Jonathan B. Conroy*: Summer Scholars Program at Tufts University.

Csaba D. Tóth: Research supported in part by NSF DMS-0701280.

Acknowledgements We thank Sujoy Bore for helpful discussions on geometric intersections graphs.

1 Introduction

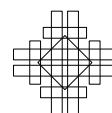
Graph spanners were introduced by Awerbuch [7] and by Peleg and Schäffer [54]. A spanner of a graph G is a spanning subgraph H with bounded distortion between graph distances in G and H . For an edge-weighted graph $G = (V, E)$, a spanning subgraph H is a t -spanner if $d_H(u, v) \leq t \cdot d_G(u, v)$ for all $u, v \in V$, where d_H and d_G are the shortest-path distances in H and G , respectively. The parameter $t \geq 1$ is the *stretch factor* of the spanner. A long line of research is devoted to finding spanners with desirable features, which minimize the number of edges, the weight, or the diameter; refer to a recent survey by Ahmed et al. [2].

In abstract graphs, all edges have unit weight. In a graph G of girth g , any proper subgraph H has stretch at least $g - 1$. In particular, a complete bipartite graph does not have any subquadratic size t -spanner for $t < 3$. The celebrated greedy spanner by Althöfer et al. [3] finds, for every n -vertex graph and parameter $t = 2k - 1$, a t -spanner with $O(n^{1+\frac{1}{k}})$ edges; and this bound matches the lower bound from the Erdős girth conjecture [31].



© Jonathan B. Conroy and Csaba D. Tóth;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 30; pp. 30:1–30:17
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Geometric Setting: Euclidean and Metric Spanners. Given a set P of n points in a metric space (M, δ) , consider the complete graph G on P where the weight of an edge uv is the distance $\delta(u, v)$. If M has doubling dimension d (e.g., Euclidean spaces of constant dimension) the greedy algorithm by Althöfer et al. [3] constructs an $(1 + \varepsilon)$ -spanner with $\varepsilon^{-O(d)}n$ edges [45]. Specifically, every set of n points in \mathbb{R}^d admits a $(1 + \varepsilon)$ -spanner with $O(\varepsilon^{-d}n)$ edges, and this bound is the best possible [45].

Gao and Zhang [38] considered data structures for approximating the *weighted* distances in *unit disk graphs* (UDG), which are intersection graphs of unit disks in \mathbb{R}^2 . Importantly, the *weight* of an edge is the Euclidean distance between the centers. They designed a well-separated pair-decomposition (WSPD) of size $O(n \log n)$ for an n -vertex UDG. For the unit ball graphs in doubling dimensions, Eppstein and Khodabandeh [30] construct $(1 + \varepsilon)$ -spanners which also have bounded degree and total weight $O(w(MST))$, generalizing earlier work in \mathbb{R}^d by Damian et al. [23]; see also [46]. Fürer and Kasiviswanathan [37] construct a $(1 + \varepsilon)$ -spanner with $O(\varepsilon^{-2}n)$ edges for the intersection graph of n disks of arbitrary radii in \mathbb{R}^2 .

Hop-Spanners for Geometric Intersection Graphs. Unit disk graphs (UDG) were the first geometric intersection graphs for which the hop distance was studied (i.e., the unweighted version), motivated by applications in wireless communication. Spanners in this setting are often called *hop-spanners* to emphasize the use of graph-theoretic distance (i.e., hop distance), as opposed to the Euclidean distance between centers.

For an n -vertex UDG G , Yan et al. [57] constructed a subgraph H with $O(n \log n)$ edges and $d_H(u, v) \leq 3d_G(u, v) + 12$, which is a 15-hop spanner. Catusse et al. [19] showed that every n -vertex UDG admits a 5-hop spanner with at most $10n$ edges (as well as a noncrossing $O(1)$ -spanner with $O(n)$ edges). Biniarz [9] improved this bound to $9n$. Dumitrescu et al. [28] recently showed that every n -vertex UDG admits a 5-hop spanner with at most $5.5n$ edges, a 3-hop spanner with at most $11n$ edges, and a 2-hop spanner with $O(n \log n)$ edges. In this paper, we improve the bound on the size of 2-hop spanners to $O(n)$, and initiate the study of minimum 2-hop spanners of other classes of geometric intersection graphs.

Our Contributions.

1. Every unit disk graph on n vertices admits a 2-hop spanner with $O(n)$ edges (Theorem 2 in Section 2). This bound is the best possible; and it generalizes to intersection graphs of translates of a convex body in the plane (shown in the full version of the paper).
2. The intersection graph of n axis-aligned fat rectangles in \mathbb{R}^2 admits a 2-hop spanner with $O(n \log n)$ edges (Theorem 15 in Section 3). This bound is the best possible: We establish a lower bound of $\Omega(n \log n)$ for the size of 2-hop spanners in the intersection graph of n homothets of any convex body in the plane (Theorem 19 in Section 4).
3. The intersection graph of n fat convex bodies in \mathbb{R}^2 admits a 3-hop spanner with $O(n \log n)$ edges (shown in the full version of the paper).

Related Previous Work. While our upper bounds are constructive, we do not attempt to minimize the number of edges in a k -spanner for a given graph. The *minimum k -spanner* problem is to find a k -spanner H of a given graph G with the minimum number of edges. This problem is NP-hard [16, 54] for all $2 \leq k \leq o(\log n)$; already for planar graphs [10, 41]. It is also hard to approximate up to a factor of $2^{(\log^{1-\varepsilon} n)/k}$, for $3 \leq \log^{1-2\varepsilon} n$ and $\varepsilon > 0$, assuming $NP \not\subseteq BPTIME(2^{\text{poly} \log(n)})$ [25]; see also [27, 29, 42]. On the positive side, Peleg

and Krtsarz [43] gave an $O(\log(m/n))$ -approximation for the minimum 2-spanner problem for graphs G with n vertices and m edges; see also [20]. There is an $O(n)$ -time algorithm for the minimum 2-spanner problem over graphs of maximum degree at most four [17].

Classical graph optimization problems (which are often hard and hard to approximate) typically admit better approximation ratios or are fixed-parameter tractable (FPT) for geometric intersection graphs. Three main strategies have been developed to take advantage of geometry: (i) Divide-and-conquer strategies using separators and dynamic programming [4, 8, 24, 18, 34, 35, 36, 47]; (ii) Local search algorithms [14, 21, 40, 51]; and (iii) Bounded VC-dimension and the ε -net theory [1, 6, 13, 53, 50, 52]. It is unclear whether separators and local search help find small k -hop spanners. Small hitting sets and ε -nets help finding large cliques in geometric intersection graphs, and this is a tool we use, as well.

Relation to Edge Clique and Biclique Covers. A 2-hop spanner H of a graph $G = (V, E)$ is union of stars \mathcal{S} such that every edge in E is induced by a star in \mathcal{S} . Thus the minimum 2-spanner problem is equivalent to minimizing the sum of sizes of stars in \mathcal{S} . As such, the 2-spanner problem is similar to the *minimum dominating set* and *minimum edge-clique cover* problems [32, 49]. In particular, the size of a 2-hop spanner is bounded above by the minimum *weighted* edge clique cover, where the weight of a clique K_t is $t - 1$ (i.e., the size of a spanning star). Recently, de Berg et al. [24] proposed a divide-and-conquer framework for optimization problems on geometric intersection graphs. Their main technical tool is a weighted separator theorem, where the weight of a separator is $W = \sum_i w(t_i)$ for a decomposition of the subgraph induced by the separator into cliques K_{t_i} , and sublinear weights $w(t) = o(t)$. For 2-hop spanners, however, each clique K_t requires a star with $t - 1$ edges, so the weight function would be linear $w(t) = t - 1$.

Every biclique (i.e., complete bipartite graph) $K_{s,t}$ admits a 3-hop spanner with $s + t - 1$ edges (as a union of two stars). Hence an *edge biclique cover*, with total weight W and weight function $w(K_{s,t}) = s + t$, yields a 3-hop spanners with at most W edges. Every n -vertex graph has an edge biclique cover of weight $O(n^2 / \log n)$, and this bound is tight [33, 56]. (In contrast, every n -vertex graph has a 3-hop spanner with $O(n^{3/2})$ edges [3].) Better bounds are known for semi-algebraic graphs, where the edges are defined in terms of semi-algebraic relations of bounded degree. For instance, an incidence graph between n points and m hyperplanes in \mathbb{R}^d admits an edge biclique cover of weight $O((mn)^{1-1/d} + m + n)$ [5, 11, 55]. Recently, Do [26] proved that a semi-algebraic bipartite graph on $m + n$ vertices, where the vertices are points in \mathbb{R}^{d_1} and \mathbb{R}^{d_2} , resp., has an edge biclique cover of weight $O_\varepsilon(m^{\frac{d_1 d_2 - d_2}{d_1 d_2 - 1} + \varepsilon} n^{\frac{d_1 d_2 - d_1}{d_1 d_2 - 1} + \varepsilon} + m^{1+\varepsilon} + n^{1+\varepsilon})$ for any $\varepsilon > 0$. For $d_1 + d_2 \leq 4$, this result yields nontrivial 3-hop spanners. For a UDG with $m = n$ unit disks, $d_1 = d_2 = 2$ gives a 3-hop spanner with $W \leq O_\varepsilon(n^{4/3+\varepsilon})$ edges. But for the intersection graph of arbitrary disks in \mathbb{R}^2 , $d_1 = d_2 = 3$ gives $O_\varepsilon(n^{3/2+\varepsilon})$, which is worse than the default $O(n^{3/2})$ guaranteed by the greedy algorithm [3].

Representation. Our algorithms assume a geometric representation of a given intersection graphs (it is NP-hard to recognize UDGs [12], disk graphs [39, 48], or box graphs [44]). Given a set of geometric objects of bounded description complexity, the intersection graph and the hop distances can easily be computed in polynomial time. Chan and Skrepetos [22] designed near-quadratic time algorithms to compute all pairwise hop-distances in the intersection graph of n geometric objects (e.g., balls or hyperrectangles in \mathbb{R}^d). In a UDG, the hop-distance between a given pair of disks can be computed in optimal $O(n \log n)$ time [15].

2 Two-Hop Spanners for Unit Disk Graphs

In this section, we prove that every n -vertex UDG has a 2-hop spanner with $O(n)$ edges. The proof hinges on a key lemma, Lemma 1, in a bipartite setting. A unit disk is a closed disk of unit diameter in \mathbb{R}^2 ; two unit disks intersect if and only if their centers are at distance at most 1 apart. For finite sets $A, B \subset \mathbb{R}^2$, let $U(A, B)$ denote the unit disk graph on $A \cup B$, and let $G(A, B)$ denote the bipartite subgraph of $U(A, B)$ of all edges between A and B .

► **Lemma 1.** *Let $P = A \cup B$ be a set of n points in the plane such that $\text{diam}(A) \leq 1$, $\text{diam}(B) \leq 1$, and A (resp., B) is above (resp., below) the x -axis. Then there is a subgraph H of $U(A, B)$ with at most $5n$ edges such that for every edge ab of $G(A, B)$, H contains a path of length at most 2 between a and b .*

We construct the graph H in Lemma 1 incrementally: In each step, we find a subset $W \subset A \cup B$, together with a subgraph $H(W)$ of at most $5|W|$ edges that contains a uv -path of length at most 2 for every edge uv between $u \in W$ and $v \in N(W)$ (cf. Lemma 5); and then recurse on $P \setminus W$. We show that $\bigcup_W H(W)$ is a 2-hop spanner for $U(A, B)$.

Section 2.1 establishes a technical lemma about the interaction pattern of disks in the bipartite setting. One step of the recursion is presented in Section 2.2. The proof of Lemma 1 is in Section 2.3. Lemma 1, combined with previous work [9, 19, 28] that reduced the problem to a bipartite setting, implies the main result of this section.

► **Theorem 2.** *Every n -vertex unit disk graph has a 2-hop spanner with $O(n)$ edges.*

Proof. Let P be a set of centers of n unit disks in the plane, and let $G = (P, E)$ be the UDG on P . Consider a tiling of the plane with regular hexagons of diameter 1, where each point in P lies in the interior of a tile. A tile τ is *nonempty* if $\tau \cap P \neq \emptyset$. Clearly $\text{diam}(P \cap \tau) \leq \text{diam}(\tau) = 1$. For each nonempty tile τ , let S_τ be a spanning star on $P \cap \tau$.

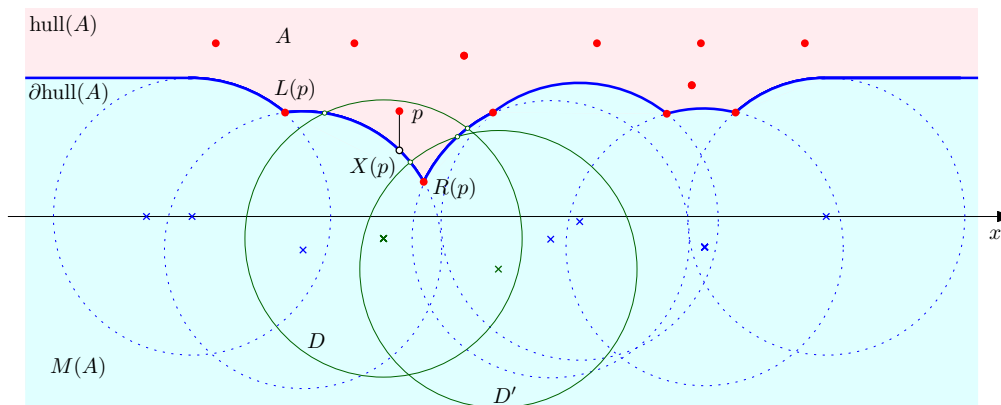
For each pair of tiles, σ and τ , at distance at most 1 apart, Lemma 1 yields a graph $H_{\sigma, \tau} := G(A, B) \subset G$ for $A = P \cap \sigma$ and $B = P \cap \tau$ with $5(|P \cap \sigma| + |P \cap \tau|)$ edges. Let H be the union of all stars S_τ and all graphs $H_{\sigma, \tau}$. It is easily checked that H is a 2-hop spanner of G : Indeed, let $uv \in E$. If u and v are in the same tile τ , then S_τ contains uv or a uv -path of length 2. Otherwise u and v are in different tiles, say σ and τ , at distance at most 1, and $H_{\sigma, \tau}$ contains uv or a uv -path of length 2.

It remains to bound the number of edges in H . The union of all stars S_τ is a spanning forest on P , which has at most $n - 1$ edges. Every tile σ is within unit distance from 18 other tiles [9]. The total number of edges in $H_{\sigma, \tau}$ over all pairs of tiles is $\sum_{\sigma, \tau} 5(|P \cap \sigma| + |P \cap \tau|) \leq 18 \sum_{\sigma} 5(|P \cap \sigma|) = 90n$. Overall, H has less than $91n$ edges, as required. ◀

2.1 Properties of Unit-Disk Hulls

Let $A \subset \mathbb{R}^2$ be a finite set of points above the x -axis. Let \mathcal{D} be the set of all unit disks with centers on or below the x -axis. Let $M(A)$ be the union of all unit disks $D \in \mathcal{D}$ such that $A \cap \text{int}(D) = \emptyset$, and let $\text{hull}(A) = \mathbb{R}^2 \setminus \text{int}(M(A))$; see Fig. 1.

For every $p \in \mathbb{R}^2$ above the x -axis, let $X(p)$ denote its vertical projection onto $\partial \text{hull}(A)$; this is well defined by Lemma 3(1) below. Let $L(p)$ and $R(p)$ denote the points in $A \cap \partial \text{hull}(A)$ immediately to the left and right of $X(p)$ if such a point exists; that is, $L(p)$ (resp., $R(p)$) is the point in $A \cap \partial \text{hull}(A)$ with the largest (resp., smallest) x -coordinate that still satisfies $L(p)_x \leq X(p)_x$ (resp., $R(p)_x \geq X(p)_x$).



■ **Figure 1** A point set A (red), region $M(A)$ (light blue), and $\text{hull}(A)$ (pink). A point $p \in A$ in a disk $D \in \mathcal{D}$, its vertical projection $X(p) \in \partial\text{hull}(A)$, and the two adjacent points $L(p), R(p) \in A$.

► **Lemma 3.** For every finite set $A \subset \mathbb{R}^2$ above the x -axis, the following holds:

1. $\partial\text{hull}(A)$ is an x -monotone curve.
2. For every $D \in \mathcal{D}$, the intersection $D \cap \partial\text{hull}(A)$ is connected (possibly empty).
3. For every $D \in \mathcal{D}$ and every $p \in A$, if $p \in D$, then D contains $X(p)$. Further, $L(p)$ or $R(p)$ exists, and D contains $L(p)$ or $R(p)$ (possibly both).
4. Let $D, D' \in \mathcal{D}$. Suppose that ∂D intersects $\partial\text{hull}(A)$ at points with x -coordinates x_1 and x_2 , and $\partial D'$ intersects $\partial\text{hull}(A)$ at points with x -coordinates x'_1 and x'_2 . If $x_1 \leq x'_1 \leq x'_2 \leq x_2$, then $D' \cap \text{hull}(A) \subset D \cap \text{hull}(A)$.

The proof (in the full version of the paper) is a straightforward extension of previous results [28, Lemma 4].

2.2 One Incremental Step

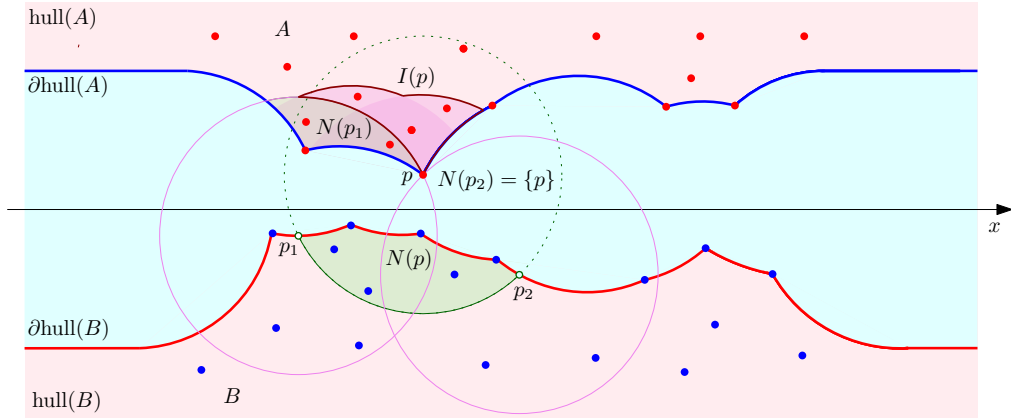
Let A and B be finite point sets above and below the x -axis, respectively, and let $P = A \cup B$. For every point $p \in \mathbb{R}^2$, let $N(p) \subset P$ denote the points in P on the opposite side of the x -axis within unit distance from p ; refer to Fig. 2. For a point set $S \subset \mathbb{R}^2$, let $N(S) = \bigcup_{p \in S} N(p)$. Suppose that a unit circle centered at $p \in A$ intersects $\partial\text{hull}(B)$ at points $p_1, p_2 \in \mathbb{R}^2$; or a unit circle centered at $p \in B$ intersects $\partial\text{hull}(A)$ at points $p_1, p_2 \in \mathbb{R}^2$. Define $I(p) = N(N(p)) \setminus (N(p_1) \cup N(p_2))$; see Fig. 2 for an example.

► **Lemma 4.** Let $P = A \cup B$ be a finite set of points in the plane such that A (resp., B) is above (resp., below) the x -axis. For every $p \in P$, $N(I(p)) \subset N(p)$.

Proof. We may assume w.l.o.g. that $p \in A$. Let $v \in I(p)$, and let D_v (resp., D_p) denote the unit disk centered on v (resp., p). As $v \in N(N(p))$, D_v contains some point $u \in N(p)$. Clearly, D_p contains u . By Lemma 3(3), D_v and D_p contain $X(u)$. As $D_p \cap \text{hull}(B)$ has endpoints p_1 and p_2 , Lemma 3(1)–(2) implies that $X(u)$ has x -coordinate between p_1 and p_2 . By definition of $I(p)$, $D_v \cap \text{hull}(B)$ does not contain either p_1 or p_2 , so it only contains points between p_1 and p_2 . By Lemma 3(4), $N(v) \subset N(p)$. ◀

We construct a spanner by repeatedly applying the following lemma:

► **Lemma 5.** Let $P = A \cup B$ be a set of n points in the plane such that $\text{diam}(A) \leq 1$, $\text{diam}(B) \leq 1$, and A (resp., B) is above (resp., below) the x -axis. Then there exists a nonempty subset $W \subset P$ and a graph $H(W)$ with the following properties:



■ **Figure 2** A point $p \in A$ and its neighbors $N(p) \subset B$. The unit circle centered at p intersecting $\partial\text{hull}(B)$ at p_1 and p_2 . The sets $N(p_1)$, $N(p_2)$, and $I(p)$.

1. $H(W)$ is a subgraph of $U(A, B)$;
2. $H(W)$ contains at most $5|W|$ edges;
3. for every edge ab in the neighborhood of W in $G(A, B)$, $H(W)$ contains an ab -path of length at most 2.

Proof. Let $m \in \mathbb{R}^2$ be the point that maximizes $|N(m)|$ (breaking ties arbitrarily) and let $k = |N(m)|$. Notice that m might not be in P . By Lemma 3(3), every point in $N(m)$ is within unit distance of $L(m)$ or $R(m)$; and $L(m), R(m) \in P$. Thus there exists a point $v \in P$ such that $|N(v)| \geq k/2$.

Now let $p \in P$ be the point that maximizes $|N(p)|$; and note that $|N(p)| \geq k/2$. Let $W = N(p) \cup I(p) \cup \{p\}$. Let $H(W)$ be the spanning star centered at p connected to all points in $N(N(p))$ and to all points in $N(p)$. We verify that $H(W)$ has the required properties:

1. Every point in $N(p)$ is within unit distance of p . As $p \in A$ and $N(N(p)) \subset A$, every point in $N(N(p))$ is within unit distance of p . Thus $H(W)$ is a subgraph of $U(A, B)$.
2. By definition of k , $|N(p_1)| \leq k$ and $|N(p_2)| \leq k$. Thus, $|N(N(p))| \leq 2k + |I(p)|$. Further, $|W| = |N(p)| + |I(p)| \geq k/2 + |I(p)|$. Thus $|N(N(p))| \leq 4|W|$. The spanning star $H(W)$ has $|N(N(p))| + |N(p)| - 1$ edges, so it has at most $5|W|$ edges.
3. For every $v \in N(p)$, all neighbors of v are in $N(N(p))$ by the definition of $N(\cdot)$, so the spanning star contains a path of length at most 2 to each neighbor. For every $v \in I(p) \cup \{p\}$, all neighbors of v are in $N(p)$ by Lemma 4, so the spanning star contains a path of length at most 2 to each neighbor. ◀

2.3 Proof of Lemma 1

We can now construct a sparse 2-hop spanner in the bipartite setting. We restate Lemma 1.

► **Lemma 1.** *Let $P = A \cup B$ be a set of n points in the plane such that $\text{diam}(A) \leq 1$, $\text{diam}(B) \leq 1$, and A (resp., B) is above (resp., below) the x -axis. Then there is a subgraph H of $U(A, B)$ with at most $5n$ edges such that for every edge ab of $G(A, B)$, H contains a path of length at most 2 between a and b .*

Proof. Apply Lemma 5 to find a subset $W \subset P$ and a subgraph $H(W)$. Let H be the union of $H(W)$ and the spanner constructed by recursing on $P \setminus W$. Since H is the union of subgraphs of $U(A, B)$, it is itself a subgraph of $U(A, B)$.

Stretch analysis. Suppose $a \in A$ and $b \in B$ are neighbors in $G(A, B)$. We assume w.l.o.g. that a was removed before or at the same time as b during the construction of H as part of some subset W . Then H includes a subgraph $H(W)$ that, by construction, connects a to all neighbors that have not yet been removed (including b) by paths of length at most 2.

Sparsity analysis. Each subgraph $H(W)$ in H is responsible for removing some set of points W and has at most $5|W|$ edges. Charge 5 edges to each of the $|W|$ points removed. As each point is removed exactly once, H contains at most $5n$ edges. ◀

3 Two-Hop Spanners for Axis-Aligned Squares

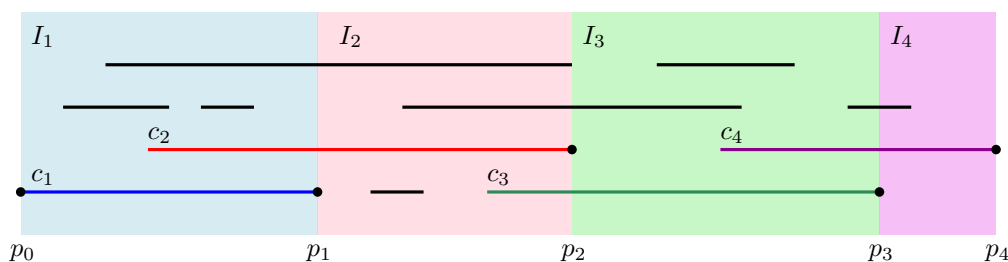
For intersection graphs of n unit disks, we found 2-hop spanners with $O(n)$ edges in Section 2. This bound does not generalize to intersection graphs of disks of arbitrary radii, as we establish a lower bound of $\Omega(n \log n)$ in Section 4. Here, we construct 2-hop spanners with $O(n \log n)$ edges for such graphs under the L_∞ norm (where unit disks are really unit squares). The result also holds for axis-aligned fat rectangles.

We prove a linear upper bound for the 1-dimensional version of the problem (Section 3.1), and then address axis-aligned fat rectangles in the plane (Section 3.2). The *fatness* of a set $s \subset \mathbb{R}^2$ is the ratio $\rho_{\text{out}}/\rho_{\text{in}}$ between the radii of a minimum enclosing disk and a maximum inscribed disk of s . A collection S of geometric objects is α -fat if the fatness of every $s \in S$ is at most α ; and it is *fat*, for short, if it is α -fat for some $\alpha \in O(1)$.

3.1 Two-Hop Spanners for Interval Graphs

Let $G(S)$ be the intersection graph of a set S of n closed segments in \mathbb{R} . Assume w.l.o.g. that $G(S)$ is connected: otherwise, we can apply this construction to each connected component.

We partition $\bigcup S$ into a collection of disjoint intervals $\mathcal{I} = \{I_1, \dots, I_m\}$ as follows. Let $I_0 = \{p_0\}$ be the interval containing only the leftmost point in $\bigcup S$, and let $k := 1$. While p_{k-1} lies to the left of the rightmost point in $\bigcup S$, let p_k be the rightmost point of any segment in S that intersects p_{k-1} ; let $I_k = (p_{k-1}, p_k]$; and set $k := k + 1$. As $G(S)$ is connected, this process terminates. For every $k \in \{1, \dots, m\}$, define the *covering segment* c_k to be some segment that intersects p_{k-1} and has right endpoint p_k ; see Fig. 3. Notice that by construction of I_k , c_k is guaranteed to exist, and $I_k \subset c_k$.



■ **Figure 3** A set of segments S , with $\bigcup S$ partitioned into intervals $\mathcal{I} = \{I_1, \dots, I_4\}$. Each $I_k \in \mathcal{I}$ is contained in some covering segment $c_k \in S$.

► **Lemma 6.** *The set of intervals \mathcal{I} defined above has the following properties:*

1. \mathcal{I} is a partition of $\bigcup S$;
2. every segment $s \in S$ intersects at most 2 intervals in \mathcal{I} ;
3. if two segments $a, b \subset \bigcup S$ intersect (with a, b not necessarily elements of S), then there is some interval in \mathcal{I} that intersects both segments.

The proof is straightforward; see the full version of the paper.

► **Theorem 7.** *Every n -vertex interval graph admits a 2-hop spanner with at most $2n$ edges.*

Proof. We construct the 2-hop spanner H as the union of stars. For every interval $I_k \in I$, construct a star H_k centered on the covering segment c_k with an edge to every segment that intersects I_k . As $I_k \subset c_k$, every segment that intersects I_k also intersects c_k , so there is an edge between the two segments in $G(S)$. Define $H = \bigcup_{k=1}^m H_k$.

Stretch analysis. Suppose $s_1, s_2 \in S$ intersect. By Lemma 6(3), $s_1 \cap s_2$ intersects some interval I_k . Thus, the star $H_k \subset H$ connects s_1 and s_2 by a path of length at most 2.

Sparsity analysis. Suppose the star $H_k \subset H$ has j edges. The corresponding interval $I_k \in I$ intersects $j + 1$ segments in S . Charge 1 edge to each of the segments intersecting I_k . By Lemma 6(2), each of the n segments in S is charged at most twice. ◀

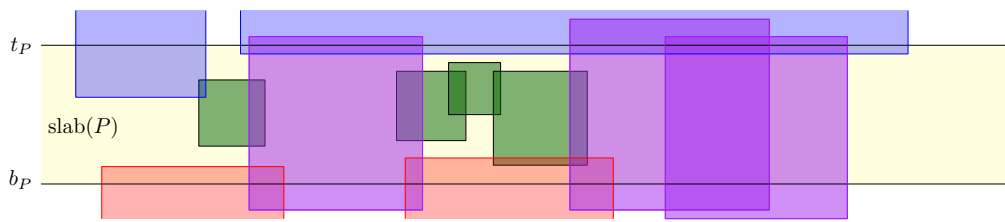
► **Corollary 8.** *The intersection graph of a set of n axis-aligned rectangles in \mathbb{R}^2 that all intersect a fixed horizontal or vertical line admits a 2-hop spanner with at most $2n$ edges.*

3.2 Two-Hop Spanners for Axis-Aligned Fat Rectangles

Let $G(S)$ be the intersection graph of a set S of n axis-aligned α -fat closed rectangles in the plane. For every pair of intersecting rectangles $a, b \in S$, select some representative point in $a \cap b$. Let $C(S)$ denote the set comprising the representatives for all intersections.

Setup for a Divide & Conquer Strategy. We recursively partition the plane into slabs by splitting along horizontal lines. The recursion tree \mathcal{P} is a binary tree, where each node $P \in \mathcal{P}$ stores a slab, denoted $\text{slab}(P)$, that is bounded by horizontal lines b_P and t_P on the bottom and top, respectively. The node P also stores a subset $S(P) \subset S$ of (not necessarily all) rectangles in S that intersect $\text{slab}(P)$.

Let the *inside set* $\text{In}(P) \subset S(P)$ be the set of rectangles contained in $\text{int}(\text{slab}(P))$. Let the *bottom set* $B(P) \subset S(P)$ be rectangles that intersect the line b_P , the *top set* $T(P) \subset S(P)$ be the rectangles that intersect t_P , and the *across set* $A(P) = B(P) \cap T(P)$; see Fig. 4.



■ **Figure 4** A horizontal slab $\text{slab}(P)$ is bounded by b_P and t_P . Rectangles in the inside set $\text{In}(P)$ (green), bottom set $B(P)$ (red), top set $T(P)$ (blue), and across set $A(P) = B(P) \cap T(P)$ (purple). Some red and blue fat rectangles are shown only partially, in a small neighborhood of $\text{slab}(P)$.

We define the root node P_r to have a slab large enough to contain all rectangles in S , and define $S(P_r) = S$. We define the rest of the space partition tree recursively. Let $P \in \mathcal{P}$. Define $C(P) \subset C(S)$ to be the set $C(S) \cap \text{int}(\text{slab}(P))$. If $C(P) = \emptyset$, then P is a leaf and has no children. Otherwise, P has two children P_1 and P_2 . Let c_P be a horizontal line with at most half the points in $C(P)$ on either side. Let $\text{slab}(P_1)$ (resp., $\text{slab}(P_2)$) be the slab bounded by b_P and c_P (resp., c_P and t_P); and let $S(P_1) \subset S(P) \setminus A(P)$ (resp.,

$S(P_2) \subset S(P) \setminus A(P)$) be the set of rectangles that intersect this slab, excluding rectangles in $A(P)$. Notice that no rectangles in $A(P)$ appear in the children of P , whereas rectangles in the sets $\text{In}(P)$, $B(P) \setminus A(P)$, and $T(P) \setminus A(P)$ appear in one or both of the children.

Spanner Construction. We construct a spanner $H(S)$ for $G(S)$ as the union of subgraphs $H(P)$ for each node P in the space partition tree.

We construct $H(P)$ such that there is a path of length at most 2 between every rectangle $s \in A(P)$ and every rectangle in $S(P)$ that s intersects. Every edge in $G(S(P))$ requiring such a path involves a rectangle in $B(P)$, a rectangle in $T(P)$, or a rectangle in $\text{In}(P)$. We construct three subgraphs to deal with these three categories of edges.

By Corollary 8, we can construct a subgraph $H_B(P)$ of $G(B(P))$ with at most $2|B(P)|$ edges that is a 2-hop spanner for $B(P)$. Similarly, we can construct a 2-hop spanner $H_T(P)$ for $G(T(P))$ with $2|T(P)|$ edges. As all rectangles in $A(S)$ intersect b_P , we can apply Corollary 8 to construct a 2-hop spanner $H'_{\text{In}}(P)$ with at most $2|A(P)|$ edges for $G(A(P))$. To construct $H_{\text{In}}(P)$, we partition $\bigcup (A(P) \cap \text{slab}(P))$ analogously to the 1-dimensional case.

Recall that by Lemma 6(1), the line segment $\bigcup (A(P) \cap b_P)$ can be partitioned into intervals I_k , each of which is contained in some covering segment $c_k \in A(P)$. As every $s \in A(P)$ is an axis-aligned rectangle that spans two horizontal lines b_P and t_P , the segments in I_k can be extended upward to form axis-aligned rectangles \widehat{I}_k , each with an associated covering rectangle $\widehat{c}_k \in S$ corresponding to the covering segment c_k in the 1-dimensional case. Let $\widehat{\mathcal{I}}$ denote the set of all 2-dimensional intervals \widehat{I}_k .

We construct $H_{\text{In}}(P)$ from $H'_{\text{In}}(P)$ using these intervals. For every $s \in \text{In}(P)$, if s intersects some $\widehat{I}_k \in \widehat{\mathcal{I}}$, add an edge between s and \widehat{c}_k to $H'_{\text{In}}(P)$. Let $H(P) = H_B(P) \cup H_T(P) \cup H_{\text{In}}(P)$.

Stretch and Weight Analysis. We start with a technical lemma (Lemma 9), which is used in the stretch and weight analysis for the graph $H(P)$ of a single node $P \in \mathcal{P}$ (Lemma 10). Notice that the intervals in $\widehat{\mathcal{I}}$ act similarly to the 1-dimensional intervals in \mathcal{I} : in particular, Lemma 6 carries over, with I_k replaced by \widehat{I}_k , and with the line segment $\bigcup S$ replaced by the region $\bigcup (A(P) \cap \text{slab}(P))$.

► **Lemma 9.** *Let w denote the smallest width of any rectangle in $S(P)$, where the width of a rectangle $s \in A(P)$ is the length of $s \cap b_P$. Then for any $k \in \mathbb{N}$, the union of any $2k$ contiguous intervals in $\widehat{\mathcal{I}}$ has width at least kw .*

Proof. By construction, every covering rectangle \widehat{c}_k intersects \widehat{I}_k and \widehat{I}_{k-1} . By Lemma 6(2), \widehat{c}_k does not intersect any other intervals in $\widehat{\mathcal{I}}$. Thus, $\widehat{c}_k \subset \widehat{I}_{k-1} \cup \widehat{I}_k$. This means that every pair of intervals has width at least w . As there are k disjoint pairs of intervals in a set containing $2k$ contiguous intervals, such a set must have width at least kw . ◀

► **Lemma 10.** *The subgraph $H(P)$ has the following properties:*

1. for every edge $ab \in G(P)$ with $a \in A(P)$, $H(P)$ contains an ab -path of length at most 2;
2. $H(P)$ contains $O(\alpha^2 |S(P)|)$ edges.

Proof.

1. Every $b \in S(P)$ is in $B(P)$, $T(P)$, or $\text{In}(P)$. If $b \in B(P)$, then the claim follows from the definition of $H_B(P)$ and the fact that $H_B(P)$ is a subgraph of $H(P)$. Similarly, the claim holds when $b \in T(P)$.

Suppose $b \in \text{In}(P)$. By Lemma 6(3), if there is an edge ab in $G(P)$ then both a and b intersect some interval \widehat{I}_k . By construction of $H_{\text{In}}(P)$, there is an edge between a and c_k and between b and c_k (or else either a or b is equal to c_k) and so there is a path of length at most 2 between a and b in $H_{\text{In}}(P)$. As $H_{\text{In}}(P)$ is a subgraph of $H(P)$, this proves the claim.

30:10 Hop-Spanners for Geometric Intersection Graphs

2. By construction, $H_B(P)$ contains $2|B(P)|$ edges, $H_T(P)$ contains $2|T(P)|$ edges, and $H_{\text{In}}^i(P)$ contains $2|A(P)|$ edges.

We now bound the number of edges that are added to $H_{\text{In}}^i(P)$ to produce $H_{\text{In}}(P)$. Let h be the distance between b_P and t_P . Every rectangle $a \in A(P)$ has width at least $\Omega(\frac{h}{\alpha})$, as a is α -fat and has height at least h . Further, notice that every rectangle $b \in \text{In}(P)$ has width less than αh , as otherwise it would cross b_P or t_P .

Let $\widehat{I}_l, \widehat{I}_r \in \widehat{I}$, resp., be the leftmost and rightmost intervals that b intersects. As these intervals are interior-disjoint, the intervals between \widehat{I}_l and \widehat{I}_r (if any exist) must have a total length less than αh ; otherwise, b could not intersect both. By Lemma 9, any consecutive $2\alpha^2$ intervals (all of length at least h/α) have width at least αh . Thus, b can intersect at most $2\alpha^2 - 1$ intervals other than \widehat{I}_l and \widehat{I}_r .

By construction, this implies that $b \in \text{In}(P)$ adds at most $O(\alpha^2)$ edges to H_{In}^i during the construction of H_{In} . Thus, H_{In} has at most $2|A(P)| + \alpha^2|\text{In}(P)|$ edges. As $\text{In}(P)$, $B(P)$, $T(P)$, and $A(P)$ are all subsets of $S(P)$, $H(P)$ has at most $O(\alpha^2|S(P)|)$ edges. ◀

We prove that $H(S) = \bigcup_{P \in \mathcal{P}} H(P)$ has $O(\alpha^2 n \log n)$ edges and that it is a 2-hop spanner. We begin by considering the size. While some $H(P)$ may contain many edges, we bound the total size of $H(S)$ by showing that every rectangle in S is involved in $O(\log n)$ subproblems.

► **Lemma 11.** *For every rectangle $s \in S$, the following hold:*

1. *there are $O(\log n)$ nodes $P \in \mathcal{P}$ where $s \in \text{In}(P)$;*
2. *there are $O(\log n)$ nodes $P \in \mathcal{P}$ where $s \in B(P) \setminus A(P)$; symmetrically, there are $O(\log n)$ nodes $P \in \mathcal{P}$ where $s \in T(P) \setminus A(P)$;*
3. *there are $O(\log n)$ nodes $P \in \mathcal{P}$ where $s \in A(P)$.*

Proof. Notice that for any k , the slabs of nodes at level k in the space partition tree have pairwise disjoint interiors. Since S contains n rectangles, there are at most $\binom{n}{2}$ intersections in $G(S)$. Thus, $|C(S)| \leq \binom{n}{2}$, and so the tree has $O(\log n)$ levels.

1. For every level $k \in \mathbb{N}$ in the space partition tree, there is only one node P where $s \in \text{In}(P)$. Suppose for the sake of contradiction that $s \in \text{In}(P_1)$ and $s \in \text{In}(P_2)$ with P_1 and P_2 in the same level and $P_1 \neq P_2$. By the definition of $\text{In}(\cdot)$, s is contained in $\text{slab}(P_1)$ and in $\text{slab}(P_2)$. As these slabs are disjoint, this is impossible. Summation over $O(\log n)$ levels of the recursion tree completes the proof.
2. For every level $k \in \mathbb{N}$ in the tree, consider the node P with the highest slab such that $\text{slab}(P) \cap s \neq \emptyset$. Notice that $s \in B(P)$ and $s \notin T(P)$, so $s \in B(P) \setminus A(P)$. Any other node P' in this level that s intersects lies strictly below P (as nodes within a level have pairwise disjoint slab interiors) and s is connected, so $s \in B(P')$ only if $s \in T(P')$. Thus, P is the only node in level k where $s \in B(P) \setminus A(P)$. A symmetric argument proves that there is only one P per level where $s \in T(P) \setminus A(P)$.
3. For every level $k \in \mathbb{N}$ in the tree, there are at most two nodes P such that $s \in A(P)$. Suppose for the sake of contradiction that there exist distinct P_1, P_2 , and P_3 at level k such that $s \in A(P_1) \cap A(P_2) \cap A(P_3)$. The interiors of the corresponding slabs are disjoint, so we may assume w.l.o.g. that P_1 lies below P_2 , which lies below P_3 . As s is connected, it intersects b_P and t_P for every node P between P_1 and P_3 . In particular, s must be in $A(P)$ for the sibling P of P_2 . Then s is also in $A(P')$ for the parent P' of P_2 . This is a contradiction – if s were in the A set of the parent of P_2 , it would not have been added to the set $S(P_2) \subset S(P') \setminus A(P')$ of rectangles for the child. ◀

► **Corollary 12.** *For every $s \in S$, there are $O(\log n)$ nodes $P \in \mathcal{P}$ where $s \in S(P)$.*

Proof. This follows from the fact that for every node P , $S(P)$ is the union of the four sets mentioned in Lemma 11: $S(P) = \text{In}(P) \cup (B(P) \setminus A(P)) \cup (T(P) \setminus A(P)) \cup A(P)$. ◀

► **Lemma 13.** $H(S)$ has $O(\alpha^2 n \log n)$ edges.

Proof. For every node P , $H(P)$ has $O(\alpha^2 |S(P)|)$ edges by Lemma 10. Charge $O(\alpha^2)$ edges to each rectangle in $S(P)$. By Corollary 12, each rectangle is charged at most $O(\log n)$ times, and so $H(S)$ has at most $O(\alpha^2 n \log n)$ edges. ◀

► **Lemma 14.** $H(S)$ is a 2-hop spanner for $G(S)$.

Proof. Let ab be an edge in $G(S)$. As the rectangles a and b intersect, there is some point $p \in C(S)$ that lies in $a \cap b$. Since p is not in the interior of any slab at the leaf level, a horizontal line of the space partition contains p . Assume w.l.o.g. that this line is b_P for some node P . If both a and b are present in $S(P)$, then $H_B(P)$ contains an ab -path of length at most 2. Otherwise, there is some node P' for which both a and b are in $S(P')$ but either a or b is not in the set for either child of P' . Assume w.l.o.g. that a was removed. By construction, a rectangle is removed exactly when it is in $A(P')$. By Lemma 10, $H(P')$ contains an ab -path of length at most 2. As $H(S) = \bigcup_{P \in \mathcal{P}} H(P)$, this proves that $H(S)$ contains such a path. ◀

The previous two lemmata prove the following theorem.

► **Theorem 15.** The intersection graph of every set of n axis-aligned α -fat rectangles in the plane admits a 2-hop spanner with $O(\alpha^2 n \log n)$ edges.

4 Lower Bound Constructions

In this section, we define a class of graphs for which any 2-hop spanner has at least $\Omega(n \log n)$ edges, then show that these graphs can be realized as the intersection graph of n homothets of any convex body in the plane.

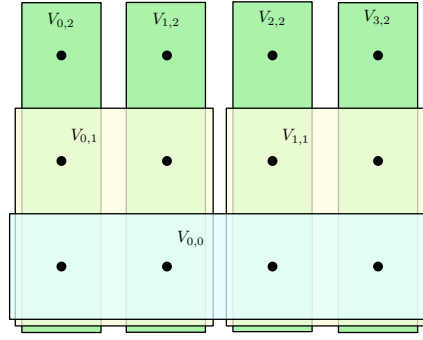
Construction of $F(h)$. For every $h \in \mathbb{N}$, we construct a graph $F(h)$, which contains $2^h(h+1)$ vertices. The vertex set is $V = \{0, \dots, 2^h - 1\} \times \{0, \dots, h\}$. For each vertex $v = (x, i)$, we call i the *level* of v . For each level $i \in \{0, \dots, h\}$, partition the vertices with level less than or equal to i into 2^i groups of $2^{h-i}(i+1)$ consecutive vertices based on their x -coordinates. In particular, for every level $i \in \{0, \dots, h\}$, let $\{0, \dots, 2^h - 1\} = \bigcup_{k=1}^{2^{i-1}} X_{k,i}$, where $X_{k,i} = \{2^{h-i}k, 2^{h-i}k + 1, \dots, 2^{h-i}(k+1) - 1\}$. This defines groups $V_{k,i} = X_{k,i} \times \{0, \dots, i\}$ for $k \in \{0, \dots, 2^i - 1\}$. Notice that $(x, \ell) \in V_{k,i}$ for $k = \lfloor x/2^i \rfloor$ and $i \geq \ell$. Finally, add edges to the graph $F(h)$ such that every group $V_{k,i}$ is a clique; see Fig. 5.

We show that any 2-hop spanner for $F(h)$ with $n = 2^h(h+1)$ vertices has $\Omega(2^h h^2) = \Omega(n \log n)$ edges. We do this by first showing that a 2-hop spanner contains $\Omega(2^{h-i} h)$ edges in each clique induced by a group $V_{k,i}$, and these edges are distinct from the edges required by any other group. This result follows from the following lemma:

► **Lemma 16.** Suppose that the vertex set of the complete graph K_{2n} is partitioned into two sets A and B each of size n , and call edges between A and B bichromatic. Then every 2-hop spanner of K_{2n} contains n bichromatic edges.

Proof. Let S be a 2-hop spanner for K_{2n} . If every vertex in A is incident to a bichromatic edge in S , then clearly S contains at least $|A| = n$ bichromatic edges. Otherwise, there is some $a \in A$ that has no direct edges to B in S . For every $b \in B$, S contains a 2-hop path between a and b , that is, a path (a, a_b, b) for some $a_b \in A$. The edges $a_b b$ are bichromatic and distinct for all $b \in B$, so S contains at least $|B| = n$ bichromatic edges. ◀

30:12 Hop-Spanners for Geometric Intersection Graphs



■ **Figure 5** Vertices of $F(2)$ grouped by cliques $V_{k,i}$.

► **Lemma 17.** For all $h \in \mathbb{N}$, $F(h)$ has $n = 2^h(h + 1)$ vertices and $\Omega(n \log n)$ edges.

Proof. Notice that every $X_{k,i}$, for $i < h$, can be written as $X_{2k,i+1} \cup X_{2k+1,i+1}$. Accordingly, we can partition $V_{k,i}$ into two sets of equal size:

$$V_{k,i} = \left(X_{2k,i+1} \times \{0, \dots, i\} \right) \cup \left(X_{2k+1,i+1} \times \{0, \dots, i\} \right).$$

Call edges that cross between these two sets $V_{k,i}$ -bichromatic.

We claim that the set of $V_{k,i}$ -bichromatic edges and $V_{k',i'}$ -bichromatic edges are disjoint unless $k' = k$ and $i' = i$. If $i = i'$, then the claim follows from the fact that $V_{k,i}$ and $V_{k',i}$ are disjoint. Otherwise, assume w.l.o.g. that $i < i'$. Notice that either $X_{k',i'}$ is contained within $X_{2k,i+1}$ or $X_{2k+1,i+1}$, or it is disjoint from both. The $V_{k,i}$ -bichromatic edges cross from $X_{2k,i+1}$ to $X_{2k+1,i+1}$ while $V_{k',i'}$ -bichromatic edges stay within $X_{k',i'}$, so the edge sets must be disjoint.

Let S be a 2-hop spanner of $F(h)$. Each vertex set $V_{k,i}$ contains $2^{h-i}(i + 1)$ vertices, so the partition described above involves two sets of size $2^{h-i-1}(i + 1)$. As $V_{k,i}$ is a clique, Lemma 16 implies that S contains at least $2^{h-i-1}(i + 1)$ $V_{k,i}$ -bichromatic edges. Every level i contains 2^i groups $V_{k,i}$, so S contains at least $2^{h-1}(i + 1)$ bichromatic edges in each level. Summation over all h levels (excluding the level where $i = h$) yields at least $\sum_{i=0}^{h-1} 2^{h-1}(i + 1) = \Omega(2^h h^2) = \Omega(n \log n)$ edges. ◀

Geometric Realization of $F(h)$. We realize $F(h)$ as the intersection graph of a set $S(h)$ of homothets of any convex body for all $h \in \mathbb{N}$. The construction is recursive. To construct $S(h+1)$, we form two copies of $S(h)$ to realize vertices in the first h levels, then add homothets to realize the vertices in level $h + 1$.

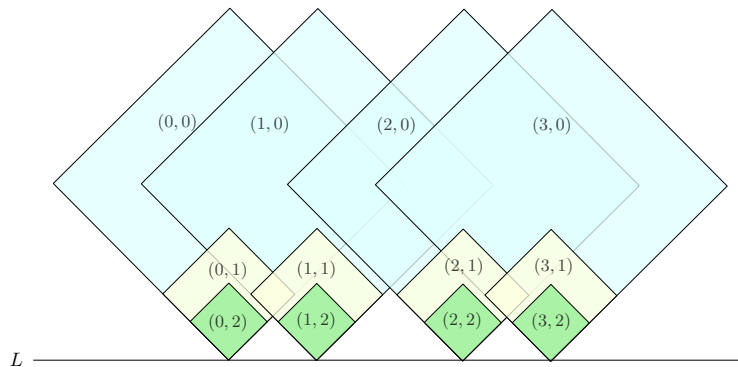
► **Lemma 18.** For every convex body $C \subset \mathbb{R}^2$ and every $h \in \mathbb{N}$, the n -vertex graph $F(h)$ can be realized as the intersection graph of a set $S(h)$ of n homothets of C .

Proof. Let C be a convex body (i.e., a compact convex set with nonempty interior) in the plane. Let $o \in \partial C$ be an extremal point of C . Then there exists a (tangent) line L such that $C \cap L = \{o\}$. Assume w.l.o.g. that o is the origin, L is the x -axis, and C lies in the upper halfplane. We construct $S(h)$ recursively from $S(h - 1)$. Let $s(a, i) \in S(h)$ denote the homothet that represents the vertex $(a, i) \in F(h)$. We maintain two invariants: (I1) for every $a \in \{0, \dots, 2^h - 1\}$, there is some point p_a on the x -axis such that every $s(a, i) \in S(h)$ is tangent to the x -axis and intersects the x -axis exactly at p_a ; and (I2) whenever $s_1, s_2 \in S(h)$ intersect, $s_1 \cap s_2$ has nonempty interior.

Construction. $F(0)$ has a single vertex $(0, 0)$ and no edges, so it can be represented as the single convex body C with the extremal point o on the x -axis.

We now construct $S(h)$ from $S(h - 1)$; see Fig. 6. By invariant (I2), there is some $\varepsilon > 0$ such that for every $s \in S(h - 1)$, translating s by ε in any direction does not change the intersection graph. Duplicate $S(h - 1)$ to form the sets $S_1(h - 1)$ and $S_2(h - 1)$, and translate every homothet in $S_2(h - 1)$ by ε in the positive x direction. Let $S'(h) = S_1(h - 1) \cup S_2(h - 1)$. Notice that for every clique $V_{k,i}$ in $F(h - 1)$, there is a corresponding clique in the intersection graph of $S'(h)$ that contains both the vertices in the clique $V_{k,i}$ realized by $S_1(h - 1)$ and the vertices in the clique $V_{k,i}$ realized by $S_2(h - 1)$.

The x -axis is still tangent to all $s \in S'(h)$, and there are 2^h distinct points on the x -axis that intersect some $s \in S'(h)$. Each point p_a has a neighborhood that intersects only the homothets in $S'(h)$ that contain p_a , since every convex body that does not contain p_a has a positive distance from p_a by compactness. For each p_a , add a homothetic copy C_a of C completely contained within that neighborhood, tangent to the x -axis and containing p_a . Let $S(h)$ be the union of $S'(h)$ and these C_a .



■ **Figure 6** Realization of $F(2)$ with homothets of squares, all tangent to L . Each homothet is labeled with the vertex of $F(2)$ that it represents.

Correctness. For $0 \leq i < h$, let the homothet $s_1(a, i) \in S_1(h - 1)$ represent $(2a, i)$ in $F(h)$, and let the homothet $s_2(a, i) \in S_2(h - 1)$ represent $(2a + 1, i)$ in $F(h)$. Let the homothets C_a represent $(a, h) \in F(h)$.

This correspondence implies that, for all $0 \leq i < h$, vertices in the clique $V_{k,i}$ in $F(h)$ have been realized by homothets corresponding to a clique $V_{\lfloor \frac{k}{2} \rfloor, i}$ in $S_1(h - 1)$ or $S_2(h - 1)$. By construction of $S(h)$, any two such homothets intersect. Similar reasoning applies in the opposite direction: any intersection between two homothets in S' corresponds to an edge in some clique in $F(h)$. When $i = h$, notice that every C_a intersects exactly the homothets in $S'(h)$ that intersect p_a , which by assumption were the homothets representing points with the same x -coordinate. Thus, any clique $V_{k,h}$ in $F(h)$ is represented in $S(h)$, and there are no edges involving C_a that do not correspond to such a clique in $F(h)$. ◀

The previous two lemmata imply the following theorem.

► **Theorem 19.** For every convex body $C \subset \mathbb{R}^2$, there exists a set S of n homothets of C such that every 2-hop spanner for the intersection graph of S has $\Omega(n \log n)$ edges.

5 Outlook

We have shown that every n -vertex UDG admits a 2-hop spanner with $O(n)$ edges; and this bound generalizes to the intersection graphs of translates of any convex body in the plane (see the full paper). The proof crucially relies on new results on the α -hull of a planar point set. It remains an open problem whether these results generalize to higher dimensions, and whether unit ball graphs admit 2-hop spanners with $O_d(n)$ edges in \mathbb{R}^d for any $d \geq 3$.

We proved that the intersection graph of n axis-aligned squares in \mathbb{R}^2 admits a 2-hop spanner with $O(n \log n)$ edges, and this bound is the best possible. However, it is unclear whether the upper bound generalizes to Euclidean disks of arbitrary radii (or to fat convex bodies) in the plane. For fat convex bodies and for axis-aligned rectangles, we obtained 3-hop spanners with $O(n \log n)$ and $O(n \log^2 n)$ edges, respectively. However, it is unclear whether the logarithmic factors are necessary. Do these intersection graphs admit weighted edge biclique covers of weight $O(n)$? In general, we do not even know whether a linear bound can be established for any constant stretch: Is there a constant $t \in \mathbb{N}$ for which every intersection graph of n disks or rectangles admits t -hop spanner with $O(n)$ edges?

Finally, it would be interesting to see other classes of intersection graphs (e.g., for strings or convex sets in \mathbb{R}^2 , set systems with bounded VC-dimension or semi-algebraic sets in \mathbb{R}^d) for which the general bound of $O(n^{1+1/\lceil t/2 \rceil})$ edges for t -hop spanners can be improved.

References

- 1 Pankaj K. Agarwal and Jiangwei Pan. Near-linear algorithms for geometric hitting sets and set covers. *Discret. Comput. Geom.*, 63(2):460–482, 2020. doi:10.1007/s00454-019-00099-6.
- 2 Abu Reyan Ahmed, Greg Bodwin, Faryad Darabi Sahneh, Keaton Hamm, Mohammad Javad Latifi Jebelli, Stephen G. Kobourov, and Richard Spence. Graph spanners: A tutorial review. *Comput. Sci. Rev.*, 37:100253, 2020. doi:10.1016/j.cosrev.2020.100253.
- 3 Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993. doi:10.1007/BF02189308.
- 4 Shinwoo An and Eunjin Oh. Feedback vertex set on geometric intersection graphs. In Hee-Kap Ahn and Kunihiko Sadakane, editors, *32nd International Symposium on Algorithms and Computation, (ISAAC)*, volume 212 of *LIPICs*, pages 47:1–47:12. Schloss Dagstuhl, 2021. doi:10.4230/LIPICs.ISAAC.2021.47.
- 5 Roel Apfelbaum and Micha Sharir. Large complete bipartite subgraphs in incidence graphs of points and hyperplanes. *SIAM J. Discret. Math.*, 21(3):707–725, 2007. doi:10.1137/050641375.
- 6 Boris Aronov, Esther Ezra, and Micha Sharir. Small-size ε -nets for axis-parallel rectangles and boxes. *SIAM J. Comput.*, 39(7):3248–3282, 2010. doi:10.1137/090762968.
- 7 Baruch Awerbuch. Communication-time trade-offs in network synchronization. In *Proc. 4th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 272–276, 1985. doi:10.1145/323596.323621.
- 8 Julien Baste and Dimitrios M. Thilikos. Contraction-bidimensionality of geometric intersection graphs. In *Proc. 12th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 89 of *LIPICs*, pages 5:1–5:13. Schloss Dagstuhl, 2017. doi:10.4230/LIPICs.IPEC.2017.5.
- 9 Ahmad Biniaz. Plane hop spanners for unit disk graphs: Simpler and better. *Comput. Geom.*, 89:101622, 2020. doi:10.1016/j.comgeo.2020.101622.
- 10 Ulrik Brandes and Dagmar Handke. NP-completeness results for minimum planar spanners. In Rolf H. Möhring, editor, *Proc. 23rd Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 1335 of *LNCS*, pages 85–99. Springer, 1997. doi:10.1007/BFb0024490.

- 11 Peter Braß and Christian Knauer. On counting point-hyperplane incidences. *Comput. Geom.*, 25(1-2):13–20, 2003. doi:10.1016/S0925-7721(02)00127-X.
- 12 Heinz Breu and David G. Kirkpatrick. Unit disk graph recognition is np-hard. *Comput. Geom.*, 9(1-2):3–24, 1998. doi:10.1016/S0925-7721(97)00014-X.
- 13 Norbert Bus, Shashwat Garg, Nabil H. Mustafa, and Saurabh Ray. Tighter estimates for ϵ -nets for disks. *Comput. Geom.*, 53:27–35, 2016. doi:10.1016/j.comgeo.2015.12.002.
- 14 Norbert Bus, Shashwat Garg, Nabil H. Mustafa, and Saurabh Ray. Limits of local search: Quality and efficiency. *Discret. Comput. Geom.*, 57(3):607–624, 2017. doi:10.1007/s00454-016-9819-x.
- 15 Sergio Cabello and Miha Jejcic. Shortest paths in intersection graphs of unit disks. *Comput. Geom.*, 48(4):360–367, 2015. doi:10.1016/j.comgeo.2014.12.003.
- 16 Leizhen Cai. Np-completeness of minimum spanner problems. *Discret. Appl. Math.*, 48(2):187–194, 1994. doi:10.1016/0166-218X(94)90073-6.
- 17 Leizhen Cai and J. Mark Keil. Spanners in graphs of bounded degree. *Networks*, 24(4):233–249, 1994. doi:10.1002/net.3230240406.
- 18 Jean Cardinal, John Iacono, and Grigorios Koumoutsos. Worst-case efficient dynamic geometric independent set. In *Proc. 29th European Symposium on Algorithms (ESA)*, volume 204 of *LIPIcs*, pages 25:1–25:15. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.ESA.2021.25.
- 19 Nicolas Catusse, Victor Chepoi, and Yann Vaxès. Planar hop spanners for unit disk graphs. In Christian Scheideler, editor, *Proc. 6th (ALGOSENSORS)*, volume 6451 of *LNCS*, pages 16–30. Springer, 2010. doi:10.1007/978-3-642-16988-5_2.
- 20 Keren Censor-Hillel and Michal Dory. Distributed spanner approximation. *SIAM J. Comput.*, 50(3):1103–1147, 2021. doi:10.1137/20M1312630.
- 21 Timothy M. Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discret. Comput. Geom.*, 48(2):373–392, 2012. doi:10.1007/s00454-012-9417-5.
- 22 Timothy M. Chan and Dimitrios Skrepetos. All-pairs shortest paths in geometric intersection graphs. *J. Comput. Geom.*, 10(1):27–41, 2019. doi:10.20382/jocg.v10i1a2.
- 23 Mirela Damian, Saurav Pandit, and Sriram V. Pemmaraju. Local approximation schemes for topology control. In Eric Ruppert and Dahlia Malkhi, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 208–217. ACM, 2006. doi:10.1145/1146381.1146413.
- 24 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for exponential-time-hypothesis-tight algorithms and lower bounds in geometric intersection graphs. *SIAM J. Comput.*, 49(6):1291–1331, 2020. doi:10.1137/20M1320870.
- 25 Michael Dinitz, Guy Kortsarz, and Ran Raz. Label cover instances with large girth and the hardness of approximating basic k -spanner. *ACM Trans. Algorithms*, 12(2):25:1–25:16, 2016. doi:10.1145/2818375.
- 26 Thao Do. Representation complexities of semialgebraic graphs. *SIAM J. Discret. Math.*, 33(4):1864–1877, 2019. doi:10.1137/18M1221606.
- 27 Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *Proc. 31st ACM Symposium on Theory of Computing (STOC)*, pages 750–759, 1999. doi:10.1145/301250.301447.
- 28 Adrian Dumitrescu, Anirban Ghosh, and Csaba D. Tóth. Sparse hop spanners for unit disk graphs. *Computational Geometry*, page 101808, 2021. doi:10.1016/j.comgeo.2021.101808.
- 29 Michael Elkin and David Peleg. The hardness of approximating spanner problems. *Theory Comput. Syst.*, 41(4):691–729, 2007. doi:10.1007/s00224-006-1266-2.
- 30 David Eppstein and Hadi Khodabandeh. Optimal spanners for unit ball graphs in doubling metrics. *CoRR*, abs/2106.15234, 2021. arXiv:2106.15234.

- 31 Paul Erdős. Extremal problems in graph theory. In *Theory of Graphs and its Applications (Proc. Sympos. Smolenice, 1963)*, pages 29–36, Prague, 1964. Publishing House of the Czechoslovak Academy of Sciences. URL: https://old.renyi.hu/~p_erdos/1964-06.pdf.
- 32 Paul Erdős, A. W. Goodman, and Louis Pósa. The representation of a graph by set intersections. *Canadian Journal of Mathematics*, 18:106–112, 1966. doi:10.4153/CJM-1966-014-3.
- 33 Paul Erdős and László Pyber. Covering a graph by complete bipartite graphs. *Discret. Math.*, 170(1-3):249–251, 1997. doi:10.1016/S0012-365X(96)00124-0.
- 34 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Finding, hitting and packing cycles in subexponential time on unit disk graphs. *Discret. Comput. Geom.*, 62(4):879–911, 2019. doi:10.1007/s00454-018-00054-x.
- 35 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In Yuval Rabani, editor, *Proc. 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1563–1575, 2012. doi:10.1137/1.9781611973099.124.
- 36 Jacob Fox and János Pach. Applications of a new separator theorem for string graphs. *Comb. Probab. Comput.*, 23(1):66–74, 2014. doi:10.1017/S0963548313000412.
- 37 Martin Fürer and Shiva Prasad Kasiviswanathan. Spanners for geometric intersection graphs with applications. *J. Comput. Geom.*, 3(1):31–64, 2012. doi:10.20382/jocg.v3i1a3.
- 38 Jie Gao and Li Zhang. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM J. Comput.*, 35(1):151–169, 2005. doi:10.1137/S0097539703436357.
- 39 Petr Hliněný and Jan Kratochvíl. Representing graphs by disks and balls (a survey of recognition-complexity results). *Discret. Math.*, 229(1-3):101–124, 2001. doi:10.1016/S0012-365X(00)00204-1.
- 40 Bruno Jartoux and Nabil H. Mustafa. Optimality of geometric local search. In *Proc. 34th Symposium on Computational Geometry (SoCG)*, volume 99 of *LIPICs*, pages 48:1–48:15. Schloss Dagstuhl, 2018. doi:10.4230/LIPICs.SoCG.2018.48.
- 41 Yusuke Kobayashi. Np-hardness and fixed-parameter tractability of the minimum spanner problem. *Theor. Comput. Sci.*, 746:88–97, 2018. doi:10.1016/j.tcs.2018.06.031.
- 42 Guy Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001. doi:10.1007/s00453-001-0021-y.
- 43 Guy Kortsarz and David Peleg. Generating low-degree 2-spanners. *SIAM J. Comput.*, 27(5):1438–1456, 1998. doi:10.1137/S0097539794268753.
- 44 Jan Kratochvíl. A special planar satisfiability problem and a consequence of its np-completeness. *Discret. Appl. Math.*, 52(3):233–252, 1994. doi:10.1016/0166-218X(94)90143-0.
- 45 Hung Le and Shay Solomon. Truly optimal Euclidean spanners. In *Proc. 60th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1078–1100, 2019. doi:10.1109/FOCS.2019.00069.
- 46 Hung Le and Shay Solomon. Towards a unified theory of light spanners I: fast (yet optimal) constructions. *CoRR*, abs/2106.15596, 2021. arXiv:2106.15596.
- 47 James R. Lee. Separators in region intersection graphs. In *Proc. 8th Innovations in Theoretical Computer Science (ITCS)*, volume 67 of *LIPICs*, pages 1:1–1:8. Schloss Dagstuhl, 2017. doi:10.4230/LIPICs.ITCS.2017.1.
- 48 Colin McDiarmid and Tobias Müller. Integer realizations of disk and segment graphs. *J. Comb. Theory, Ser. B*, 103(1):114–143, 2013. doi:10.1016/j.jctb.2012.09.004.
- 49 T. S. Michael and Thomas Quint. Sphericity, cubicity, and edge clique covers of graphs. *Discret. Appl. Math.*, 154(8):1309–1313, 2006. doi:10.1016/j.dam.2006.01.004.
- 50 Nabil H. Mustafa, Kunal Dutta, and Arijit Ghosh. A simple proof of optimal epsilon nets. *Combinatorica*, 38(5):1269–1277, 2018. doi:10.1007/s00493-017-3564-5.
- 51 Nabil H. Mustafa and Saurabh Ray. Improved results on geometric hitting set problems. *Discret. Comput. Geom.*, 44(4):883–895, 2010. doi:10.1007/s00454-010-9285-9.
- 52 Nabil H. Mustafa and Kasturi R. Varadarajan. Epsilon-approximations and epsilon-nets. In Jacob E. Goodman, Joseph O’Rourke, and Csaba D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 47. CRC Press, Boca Raton, FL, 3rd edition, 2017.

- 53 János Pach and Gábor Tardos. Tight lower bounds for the size of epsilon-nets. *J. AMS*, 26:645–658, 2013. doi:10.1090/S0894-0347-2012-00759-0.
- 54 David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989. doi:10.1002/jgt.3190130114.
- 55 Micha Sharir and Noam Solomon. Incidences with curves and surfaces in three dimensions, with applications to distinct and repeated distances. In *Proc. 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2456–2475, 2017. doi:10.1137/1.9781611974782.163.
- 56 Zsolt Tuza. Covering of graphs by complete bipartite subgraphs; complexity of 0-1 matrices. *Comb.*, 4(1):111–116, 1984. doi:10.1007/BF02579163.
- 57 Chenyu Yan, Yang Xiang, and Feodor F. Dragan. Compact and low delay routing labeling scheme for unit disk graphs. *Comput. Geom.*, 45(7):305–325, 2012. doi:10.1016/j.comgeo.2012.01.015.

Persistent Cup-Length

Marco Contessoto  

Department of Mathematics, São Paulo State University – UNESP, Brazil

Facundo Mémoli  

Department of Mathematics and Department of Computer Science and Engineering,
The Ohio State University, Columbus, OH, US

Anastasios Stefanou  

Department of Mathematics and Computer Science, University of Bremen, Germany

Ling Zhou   

Department of Mathematics, The Ohio State University, Columbus, OH, US

Abstract

Cohomological ideas have recently been injected into persistent homology and have for example been used for accelerating the calculation of persistence diagrams by the software Ripser.

The cup product operation which is available at cohomology level gives rise to a graded ring structure that extends the usual vector space structure and is therefore able to extract and encode additional rich information. The maximum number of cocycles having non-zero cup product yields an invariant, the cup-length, which is useful for discriminating spaces.

In this paper, we lift the cup-length into the persistent cup-length function for the purpose of capturing ring-theoretic information about the evolution of the cohomology (ring) structure across a filtration. We show that the persistent cup-length function can be computed from a family of representative cocycles and devise a polynomial time algorithm for its computation. We furthermore show that this invariant is stable under suitable interleaving-type distances.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Theory of computation → Computational geometry; Mathematics of computing → Topology

Keywords and phrases cohomology, cup product, persistence, cup length, Gromov-Hausdorff distance

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.31

Related Version *Full Version:* <https://arxiv.org/abs/2107.01553> [11]

Funding *Marco Contessoto:* MC was supported by FAPESP through grants 2016/24707-4, 2017/25675-1 and 2019/22023-9.

Facundo Mémoli: FM was partially supported by the NSF through grants RI-1901360, CCF-1740761, and CCF-1526513, and DMS-1723003.

Anastasios Stefanou: AS was supported by NSF through grants CCF-1740761, DMS-1440386, RI-1901360, and the Dioscuri program initiated by the Max Planck Society, jointly managed with the National Science Centre (Poland), and mutually funded by the Polish Ministry of Science and Higher Education and the German Federal Ministry of Education and Research.

Ling Zhou: LZ was partially supported by the NSF through grants RI-1901360, CCF-1740761, and CCF-1526513, and DMS-1723003.

1 Introduction

Persistent Homology [20, 21, 33, 40, 10, 18, 8, 9], one of the main techniques in *Topological Data Analysis (TDA)*, studies the evolution of homology classes across a filtration. This produces a collection of birth-death pairs which is called the *barcode* or *persistence diagram* of the filtration.



© Marco Contessoto, Facundo Mémoli, Anastasios Stefanou, and Ling Zhou;
licensed under Creative Commons License CC-BY 4.0

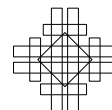
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 31; pp. 31:1–31:17

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

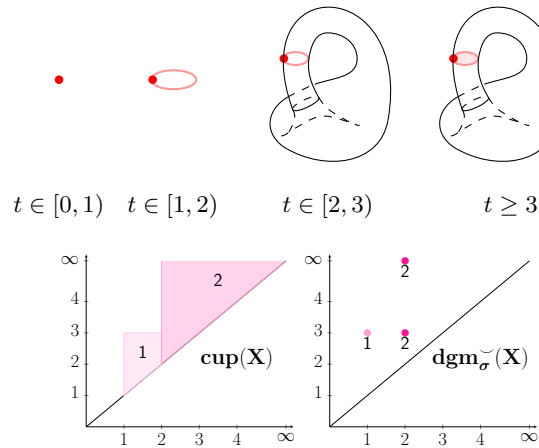


31:2 Persistent Cup-Length

In the case of *cohomology*, which is dual to that of homology, one studies linear maps from the vector space of simplicial chains into the field K , known as *cochains*. Cochains are naturally endowed with a product operation, called the *cup product*, which induces a bilinear operation on cohomology and is denoted by $\smile: \mathbf{H}^p(\mathbb{X}) \times \mathbf{H}^q(\mathbb{X}) \rightarrow \mathbf{H}^{p+q}(\mathbb{X})$ for a space \mathbb{X} and dimensions $p, q \geq 0$. With the cup product operation, the collection of cohomology vector spaces can be given the structure of a *graded ring*, called the *cohomology ring*; see [31, § 48 and § 68] and [23, Ch. 3, §3.D]. This makes cohomology a richer structure than homology.

Persistent cohomology has been studied in [15, 16, 17, 5, 27], without exploiting the ring structure induced by the cup product. Works which do attempt to exploit this ring structure include [22, 26] in the static case and [25, 39, 3, 29, 24, 6, 12] at the persistent level.

In this paper, we continue this line of work and tackle the question of quantifying the *evolution* of the cup product structure across a filtration through introducing a *polynomial-time computable* invariant which is induced from the *cup-length*: the maximal number of cocycles (in dimensions 1 and above) having non-zero cup product. We call this invariant the *persistent cup-length function*, and identify a tool - the *persistent cup-length diagram* (associated to a family of representative cocycles σ of the barcode) to compute it. (see Fig. 1).



■ **Figure 1** A filtration \mathbf{X} of the pinched Klein bottle, its persistent cup-length function $\mathbf{cup}(\mathbf{X})$ (see Ex.12) and its persistent cup-length diagram $\mathbf{dgm}_{\sigma}^{\smile}(\mathbf{X})$ (see Ex. 17).

Some invariants related to the cup product. In standard topology, an *invariant* is a quantity associated to a given topological space which remains invariant under a certain class of maps. This invariance helps in discovering, studying and classifying properties of spaces. Beyond *Betti numbers*, examples of classical invariants are: the *Lusternik-Schnirelmann category* (*LS-category*) of a space \mathbb{X} , defined as the minimal integer $k \geq 1$ such that there is an open cover $\{U_i\}_{i=1}^k$ of \mathbb{X} such that each inclusion map $U_i \hookrightarrow \mathbb{X}$ is null-homotopic, and the *cup-length invariant*, which is the maximum number of positive-dimensional cocycles having non-zero cup product. While being relatively more informative, the LS-category is difficult to compute [13], and with rational coefficients this computation is known to be NP-hard [2]. The cup-length invariant, as a lower bound of the LS-category [34, 35], serves as a computable estimate for the LS-category. Another well known invariant which can be estimated through the cup-length is the so-called *topological complexity* [38, 19, 36].

Our contributions

Let **Top** denotes the category of (compactly generated weak Hausdorff) topological spaces.¹ Throughout the paper, by a (topological) space we refer to an object in **Top**, and by a persistent space we mean a functor from the poset category (\mathbb{R}, \leq) to **Top**. A filtration (of spaces) is an example of a persistent space where the transition maps are given by inclusions. This paper considers only persistent spaces with a discrete set of critical values. In addition, all (co)homology groups are assumed to be taken over a field K . We denote by \mathbf{Int}_ω the set of intervals of type ω , where ω can be any one of the four types: open-open, open-closed, closed-open and closed-closed. The type ω will be omitted when the results apply to all four situations and intervals are written in the form of $\langle a, b \rangle$.

We introduce the invariant, the persistent cup-length function of general persistent spaces, by lifting the standard cup-length invariant into the persistent setting. Let $\mathbf{X} : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ be a persistent space with $t \mapsto \mathbb{X}_t$. The **persistent cup-length function** $\mathbf{cup}(\mathbf{X}) : \mathbf{Int} \rightarrow \mathbb{N}$ of \mathbf{X} , see Defn. 7, is defined as the function from the set \mathbf{Int} to the set \mathbb{N} of non-negative integers, which assigns to each interval $\langle a, b \rangle$ the cup-length of the image ring² $\mathbf{Im}(\mathbf{H}^*(\mathbf{X})\langle a, b \rangle)$, which is the ring $\mathbf{Im}(\mathbf{H}^*(\mathbb{X}_b) \rightarrow \mathbf{H}^*(\mathbb{X}_a))$ when $\langle a, b \rangle$ is a closed interval (in other cases, there is some subtlety, see Rmk. 8). Note that the persistent cup-length function is a generalization of the cup-length of spaces, since $\mathbf{cup}(\mathbf{X})([a, a])$ reduces to the cup-length of the space \mathbb{X}_a .

In the case when \mathbf{X} is a filtration, we define a notion of a diagram to compute the persistent cup-length function (see Thm. 1): the **persistent cup-length diagram** $\mathbf{dgm}_\sigma^\sim(\mathbf{X}) : \mathbf{Int} \rightarrow \mathbb{N}$ (Defn. 16). We first assign a representative cocycle to every interval in the barcode of \mathbf{X} , and denote the family of representative cocycles by σ . Then, the persistent cup-length diagram of an interval $\langle a, b \rangle$ is defined to be the maximum number of representative cocycles in \mathbf{X} that have a nonzero cup product over $\langle a, b \rangle$. It is worth noticing that the persistent cup-length diagram depends on the choice of representative cocycles; see Ex. 18.

► **Theorem 1.** *Let \mathbf{X} be a filtration, and let σ be a family of representative cocycles for the barcodes of \mathbf{X} . The persistent cup-length function $\mathbf{cup}(\mathbf{X})$ can be retrieved from the persistent cup-length diagram $\mathbf{dgm}_\sigma^\sim(\mathbf{X})$: for any $\langle a, b \rangle \in \mathbf{Int}$,*

$$\mathbf{cup}(\mathbf{X})(\langle a, b \rangle) = \max_{\langle c, d \rangle \supseteq \langle a, b \rangle} \mathbf{dgm}_\sigma^\sim(\mathbf{X})(\langle c, d \rangle). \tag{1}$$

The persistent cup-length functions do not supersede the standard persistence diagrams, partly because they do not take \mathbf{H}^0 classes into account. However, it effectively augments the standard diagram in the sense that there are situations in which it can successfully capture information that standard persistence diagrams neglect (see Fig. 5). Our work therefore provides additional *computable* persistence-like invariants enriching the TDA toolset which can be used in applications requiring discriminating between different hypotheses such as in shape classification or machine learning. For example, [27] mentions that cup product could provide additional evidence when recovering the structure of animal trajectories.

A polynomial time algorithm. We develop a poly-time algorithm (Alg. 3) to compute the persistent cup-length diagram of a filtration \mathbf{X} of a simplicial complex \mathbb{X} of dimension $(k + 1)$. This algorithm is output sensitive, and it has complexity bounded above by

¹ We are following the convention from [7].

² For $f : R \rightarrow S$ a graded ring morphism, we denote the graded ring $f(R)$ by $\mathbf{Im}(f)$.

31:4 Persistent Cup-Length

$O((m_k)^2 \cdot q_1 \cdot q_{k-1} \cdot \max\{c_k, q_1\}) \leq O((m_k)^{k+3})$ (cf. Thm. 20), with m_k being the cardinality of \mathbb{X} , q_1 being the cardinality of the barcode, and parameters $q_{k-1} (\leq q_1^{k-1})$ and $c_k (\leq m_k)$ which we describe in §3.2 on page 13. In the case of the Vietoris-Rips filtration of an n -point metric space, this complexity is improved to $O((m_k)^2 \cdot q_1^2 \cdot q_{k-1})$, which can be upper bounded by $O(n^{k^2+5k+6})$.

Gromov-Hausdorff stability and discriminating power. In Thm. 2 we prove that the persistent cup-length function is stable to perturbations of the involved filtrations (in a suitable sense involving weak homotopy equivalences). Below, d_E, d_{HI} and d_{GH} denote the erosion, homotopy-interleaving and Gromov-Hausdorff distances, respectively. See [11, §D] for details.

In general, the Gromov-Hausdorff distance is NP-hard to compute [37] whereas the erosion distance is computable in polynomial time (see [28, Thm. 5.4]) and thus, in combination with Thm. 2, provides a computable estimate for the Gromov-Hausdorff distance.

► **Theorem 2 (Homotopical stability).** *For two persistent spaces $\mathbf{X}, \mathbf{Y} : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$,*

$$d_E(\mathbf{cup}(\mathbf{X}), \mathbf{cup}(\mathbf{Y})) \leq d_{HI}(\mathbf{X}, \mathbf{Y}). \quad (2)$$

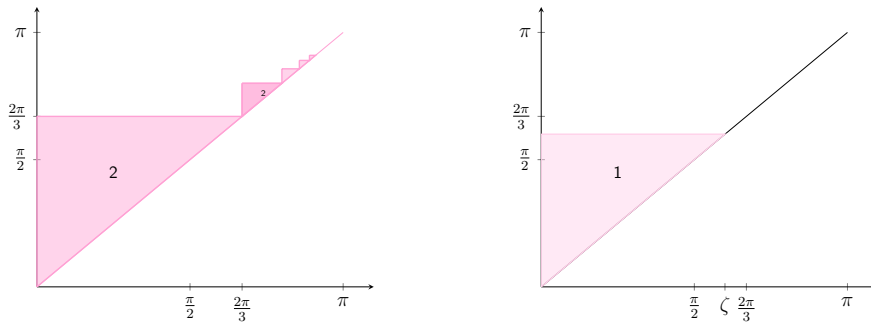
For the Vietoris-Rips filtrations $\mathbf{VR}(X)$ and $\mathbf{VR}(Y)$ of compact metric spaces X and Y ,

$$d_E(\mathbf{cup}(\mathbf{VR}(X)), \mathbf{cup}(\mathbf{VR}(Y))) \leq 2 \cdot d_{GH}(X, Y). \quad (3)$$

Through several examples, we show that the persistent cup-length function helps in discriminating filtrations when the persistent homology fails to or has a relatively weak performance in doing so. Ex. 13 is a situation when two filtrations have identical persistent homology but induce different persistent cup-length functions. In addition, in [11, Ex. 54] by specifying suitable metrics on the torus \mathbb{T}^2 and on the wedge sum $\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1$, we compute the erosion distance between their persistent cup-length functions (see Fig. 2) and apply Thm. 2 to obtain a lower bound $\frac{\pi}{3}$ for the Gromov-Hausdorff distance between them \mathbb{T}^2 and $\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1$ (see [11, Prop. 55]):

$$\frac{\pi}{3} = d_E(\mathbf{cup}(\mathbf{VR}(\mathbb{T}^2)), \mathbf{cup}(\mathbf{VR}(\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1))) \leq 2 \cdot d_{GH}(\mathbb{T}^2, \mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1).$$

We also verify that the interleaving distance between the persistent homology of these two spaces is at most $\frac{3}{5}$ of the bound obtained from persistent cup-length functions (a fact which we also establish). See [11, Rmk. 56].



■ **Figure 2** The persistent cup-length functions $\mathbf{cup}(\mathbf{VR}(\mathbb{T}^2))$ (left) and $\mathbf{cup}(\mathbf{VR}(\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1))|_{(0, \zeta)}$ (right), respectively. Here, $\zeta = \arccos(-\frac{1}{3}) \approx 0.61\pi$.

Proofs of all the theorems and results mentioned above are available in the appendix of the full version [11].

2 Persistent cup-length function

In the standard setting of persistent homology, one considers a *filtration* of spaces, i.e. a collection of spaces $\mathbf{X} = \{\mathbb{X}_t\}_{t \in \mathbb{R}}$ such that $\mathbb{X}_t \subset \mathbb{X}_s$ for all $t \leq s$, and studies the *p-th persistent homology* for any given dimension p , defined as the functor $\mathbf{H}_p(\mathbf{X}) : (\mathbb{R}, \leq) \rightarrow \mathbf{Vec}$ which sends each t to the p -th homology $\mathbf{H}_p(\mathbb{X}_t)$ of \mathbb{X}_t , see [18, 8]. Here \mathbf{Vec} denotes the category of vector spaces. The p -th persistent homology encodes the lifespans, represented by intervals, of the p -dimensional holes (p -cycles that are not p -boundaries) in \mathbf{X} . The collection $\mathcal{B}_p(\mathbf{X})$ of these intervals is called *the p-th barcode of X*, and its elements are named *bars*. The *p-th persistent cohomology* $\mathbf{H}^p(\mathbf{X})$ and its corresponding barcode are defined dually. Since persistent homology and persistent cohomology have the same barcode [15], we will denote both barcodes by $\mathcal{B}_p(\mathbf{X})$ for dimension p . We call the disjoint union $\mathcal{B}(\mathbf{X}) := \sqcup_{p \in \mathbb{N}} \mathcal{B}_p(\mathbf{X})$ *the total barcode of X*, and assume bars in $\mathcal{B}(X)$ are of the same interval type³.

By considering the *cup product* operation on cocycles, the persistent cohomology is naturally enriched with the structure of a *persistent graded ring*, which carries additional information and leads to invariants stronger than standard barcodes in cases like Ex. 13.

In §2.1 we recall the cup product operation, as well as the notion and properties of the cup-length invariant of cohomology rings. In §2.2 we lift the cup-length invariant to a persistent invariant, called the persistent cup-length function, and examine some examples that highlight its strength. Proofs and details are available in [11, §B].

2.1 Cohomology rings and the cup-length invariant

For a topological space \mathbb{X} and a dimension $p \in \mathbb{N}$, denote by $C_p(\mathbb{X})$ and $C^p(\mathbb{X})$ the spaces of singular p -chains and p -cochains, respectively. For a cocycle σ , denote by $[\sigma]$ the cohomology class of σ . If \mathbb{X} is given by the geometric realization of some simplicial complex, then we consider its simplicial cohomology, by assuming an ordering on the vertex set of \mathbb{X} and considering its simplices to be sets of ordered vertices.

Let $\mathbf{X} := \{\mathbb{X}_t\}_{t \in \mathbb{R}}$ be a filtration of topological spaces, and let $I = \langle b, d \rangle \in \mathcal{B}_p(\mathbf{X})$. If I is closed at its right end d , we denote by σ_I a cocycle in $C^p(\mathbb{X}_d)$; if not, we denote by σ_I a cocycle in $C^p(\mathbb{X}_{d-\delta})$ for sufficiently small $\delta > 0$. For any $t \leq d$, denote by $\sigma_I|_{C_p(\mathbb{X}_t)}$ the restriction of σ_I to $C_p(\mathbb{X}_t) (\subset C_p(\mathbb{X}_d))$. We introduce the notation $[\sigma_I]_t$ by defining $[\sigma_I]_t$ to be $[\sigma_I|_{C_p(\mathbb{X}_t)}]$ for $t \leq d$ and 0 for $t > d$.

► **Definition 3** (Representative cocycles). *Let $\sigma^p := \{\sigma_I\}_{I \in \mathcal{B}_p(\mathbf{X})}$ be a $\mathcal{B}_p(\mathbf{X})$ -indexed collection of p -cocycles in \mathbf{X} . The collection σ^p is called a family of representative p -cocycles for $\mathbf{H}^p(\mathbf{X})$, if for any $t \in \mathbb{R}$, the set $\{[\sigma_I]_t\}_{I \in \mathcal{B}_p(\mathbf{X})}$ forms a linear basis for $\mathbf{H}^p(\mathbb{X}_t)$. In this case, each σ_I is called a representative cocycle associated to the interval I . The disjoint union $\sigma := \sqcup_{p \in \mathbb{N}} \sigma^p$ is called a family of representative cocycles for $\mathbf{H}^*(\mathbf{X})$.*

The existence of a family of representative cocycles for $\mathbf{H}^*(\mathbf{X})$ (assuming that the filtration \mathbf{X} has finite critical values and finite-dimensional cohomology point-wise) is guaranteed by the interval decomposition theorem of point-wise finite dimensional persistence modules (see [14]) and the axiom of choice. Software programs are available to compute the total barcode and return a family of representative cocycles, such as Ripser (see [5]), Java-Plex (see [1]), Dionysus (see [16]), and Gudhi (see [30]). These cocycles are naturally equipped with the cup product operation, which we recall as follows.

³ In TDA it is often the case that bars are of a fixed interval type, usually in closed-open form [32].

Cup product. We recall the cup product operation in the setting of simplicial cohomology. Let \mathbb{X} be a simplicial complex with an ordered vertex set $\{x_1 < \dots < x_n\}$. For any non-negative integer p , we denote a p -simplex by $\alpha := [\alpha_0, \dots, \alpha_p]$ where $\alpha_0 < \dots < \alpha_p$ are ordered vertices in \mathbb{X} , and by $\alpha^* : C_p(\mathbb{X}) \rightarrow K$ the dual of α , where $\alpha^*(\alpha) = 1$ and $\alpha^*(\tau) = 0$ for any p -simplex $\tau \neq \alpha$. Here K is the base field as before, and α^* is also called a p -cosimplex. Let $\beta := [\beta_0, \dots, \beta_q]$ be a q -simplex for some integer $q \geq 0$. The *cup product* $\alpha^* \smile \beta^*$ is defined as the linear map $C_{p+q}(\mathbb{X}) \rightarrow K$ such that for any $(p+q)$ -simplex $\tau = [\tau_0, \dots, \tau_{p+q}]$,

$$\alpha^* \smile \beta^*(\tau) := \alpha^*([\tau_0, \dots, \tau_p]) \cdot \beta^*([\tau_p, \dots, \tau_{p+q}]).$$

Equivalently, we have that $\alpha^* \smile \beta^*$ is $[\alpha_0, \dots, \alpha_p, \beta_1, \dots, \beta_q]^*$ if $\alpha_p = \beta_0$, and 0 otherwise. By a *p -cochain* we mean a finite linear sum $\sigma = \sum_{j=1}^h \lambda_j \alpha^{j*}$, where each α^j is a p -simplex in \mathbb{X} and $\lambda_j \in K$. The *cup product* of a p -cochain $\sigma = \sum_{j=1}^h \lambda_j \alpha^{j*}$ and a q -cochain $\sigma' = \sum_{j'=1}^{h'} \mu_{j'} \beta^{j'*}$ is defined as $\sigma \smile \sigma' := \sum_{j,j'} \lambda_j \mu_{j'} (\alpha^{j*} \smile \beta^{j'*})$.

In our algorithms, K is taken to be \mathbb{Z}_2 and every p -simplex $\alpha = [x_{i_0}, \dots, x_{i_p}]$ is represented by the ordered list $[i_0, \dots, i_p]$. We assume a total order (e.g. the order given in [5]) on the simplices in \mathbb{X} . Since coefficients are either 0 or 1, a p -cochain can be written as $\sigma = \sum_{j=1}^h \alpha^{j*}$ for some $\alpha^j = [x_{i_0^j}, \dots, x_{i_p^j}]$ and will be represented by the list $[[i_0^1, \dots, i_p^1], \dots, [i_0^h, \dots, i_p^h]]$. We call h the size of σ . Let $\mathbb{X}_p \subset \mathbb{X}$ be the set of p -simplices. Alg. 1 computes the cup product of two cochains over \mathbb{Z}_2 .

■ **Algorithm 1** CupProduct($\sigma_1, \sigma_2, \mathbb{X}$).

Input : Two cochains σ_1 and σ_2 , and the simplicial complex \mathbb{X} .

Output : The cup product $\sigma = \sigma_1 \smile \sigma_2$, at cochain level.

```

1  $\sigma \leftarrow []$ ;
2 if  $\dim(\sigma_1) + \dim(\sigma_2) \leq \dim(\mathbb{X})$  then
3   for  $i \leq \text{size}(\sigma_1)$  and  $j \leq \text{size}(\sigma_2)$  do
4      $a \leftarrow \sigma_1(i)$  and  $b \leftarrow \sigma_2(j)$ ;
5     if  $a[\text{end}] == b[\text{first}]$  then
6        $c \leftarrow a.\text{append}(b[\text{second} : \text{end}])$ ;
7       if  $c \in \mathbb{X}_{\dim(\sigma_1) + \dim(\sigma_2)}$  then
8          $\text{Append } c \text{ to } \sigma$ ;
9 return  $\sigma$ .
```

► **Remark 4** (Complexity of Alg. 1). Let c be the complexity of checking whether a simplex is in the simplicial complex, and let $m := \text{card}(\mathbb{X})$ be the number of simplices. For \mathbb{Z}_2 -coefficients, cocycles are in one-to-one correspondence with the subsets of \mathbb{X} , so the size of a cocycle is at most m . Thus, the complexity for Alg. 1 is $O(\text{size}(\sigma_1) \cdot \text{size}(\sigma_2) \cdot c) \leq O(m^2 \cdot c)$.

Cohomology ring and cup-length. For a given space \mathbb{X} , the cup product yields a bilinear map $\smile : \mathbf{H}^p(\mathbb{X}) \times \mathbf{H}^q(\mathbb{X}) \rightarrow \mathbf{H}^{p+q}(\mathbb{X})$ of vector spaces. In particular, it turns the total cohomology vector space $\mathbf{H}^*(\mathbb{X}) := \bigoplus_{p \in \mathbb{N}} \mathbf{H}^p(\mathbb{X})$ into a graded ring $(\mathbf{H}^*(\mathbb{X}), +, \smile)$ (see [11, §B] for the explicit definition of a graded ring). The *cohomology ring map* $\mathbb{X} \mapsto \mathbf{H}^*(\mathbb{X})$ defines a contravariant functor from the category of spaces, **Top**, to the category of graded rings, **GRing** (see [23, §3.2]). To avoid the difficulty of describing and comparing ring structures in a computer, we study a computable invariant of the graded cohomology ring, called the *cup-length*. See [11, §A.2] for the general notion of *invariants*. For a category \mathcal{C} , denote by $\mathbf{Ob}(\mathcal{C})$ the set of objects in \mathcal{C} .

► **Definition 5** (Length and cup-length). The *length* of a graded ring $R = \bigoplus_{p \in \mathbb{N}} R_p$ is the largest non-negative integer ℓ such that there exist positive-dimension homogeneous elements $\eta_1, \dots, \eta_\ell \in R$ (i.e. $\eta_1, \dots, \eta_\ell \in \bigcup_{p \geq 1} R_p$) with $\eta_1 \bullet \dots \bullet \eta_\ell \neq 0$. If $\bigcup_{p \geq 1} R_p = \emptyset$, then we define the length of R to be zero. We denote the length of a graded ring R by $\text{len}(R)$, and call the following map the **length invariant**:

$$\text{len} : \text{Ob}(\mathbf{GRing}) \rightarrow \mathbb{N}, \text{ with } R \mapsto \text{len}(R).$$

When $R = (\mathbf{H}^*(\mathbb{X}), +, \smile)$ for some space \mathbb{X} , we denote $\text{cup}(\mathbb{X}) := \text{len}(\mathbf{H}^*(\mathbb{X}))$ and call it the **cup-length of \mathbb{X}** . And we call the following map the **cup-length invariant**:

$$\text{cup} : \text{Ob}(\mathbf{Top}) \rightarrow \mathbb{N}, \text{ with } X \mapsto \text{cup}(X).$$

► **Remark 6** (About the strength of the cup-length invariant). In some cases, cup-length captures more information than homology. One well-known example is given by the torus \mathbb{T}^2 v.s. the wedge sum $\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1$, where despite having the same homology groups, these two spaces have *different* cup-length. By specifying suitable metrics and considering the Vietoris-Rips filtrations of the two spaces, the strength of cup-length persists in the setting of persistence (see [11, Ex. 54]). It is also worth noticing that cup-length is not a complete invariant for graded cohomology rings. For instance, after taking the wedge sum of \mathbb{T}^2 and $\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1$ with \mathbb{T}^2 respectively, the resulted spaces $\mathbb{T}^2 \vee \mathbb{T}^2$ and $\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{T}^2$ still have different ring structures, but they have the same cup-length (since cup length takes the “maximum”).

An important fact about the cup-length is that it can be computed using a linear basis for the cohomology vector space. In [11, Prop. 36] we show that if B_p is a linear basis for $\mathbf{H}^p(\mathbb{X})$ for each $p \geq 1$ and $B := \bigcup_{p \geq 1} B_p$, then $\text{cup}(\mathbb{X}) = \sup \{ \ell \geq 1 \mid B^{\smile \ell} \neq \{0\} \}$.

2.2 Persistent cohomology rings and persistent cup-length functions

We study the persistent cohomology ring of a filtration and the associated notion of persistent cup-length invariant. We examine several examples of this persistent invariant and establish a way to visualize it in the half-plane above the diagonal. See [11, §B.2] for the proofs of our results in this section.

Filtrations of spaces are special cases of *persistent spaces*. In general, for any category \mathcal{C} , one can define the notion of a *persistent object* in \mathcal{C} , as a functor from the poset (\mathbb{R}, \leq) (viewed as a category) to the category \mathcal{C} . For instance, a functor $\mathbf{R} : (\mathbb{R}, \leq) \rightarrow \mathbf{GRing}$ is called a **persistent graded ring**. Recall the contravariant cohomology ring functor $\mathbf{H}^* : \mathbf{Top} \rightarrow \mathbf{GRing}$. Given a persistent space $\mathbf{X} : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$, the composition $\mathbf{H}^*(\mathbf{X}) : (\mathbb{R}, \leq) \rightarrow \mathbf{GRing}$ is called the **persistent cohomology ring of \mathbf{X}** . Due to the contravariance of \mathbf{H}^* , we consider only contravariant persistent graded rings in this paper.

► **Definition 7.** We define the **persistent cup-length function** of a persistent space \mathbf{X} as the function $\text{cup}(\mathbf{X}) : \mathbf{Int} \rightarrow \mathbb{N}$ given by $\langle t, s \rangle \mapsto \text{len}(\mathbf{Im}(\mathbf{H}^*(\mathbf{X}))(\langle t, s \rangle))$.

► **Remark 8** (Notation for image ring). $\mathbf{Im}(\mathbf{H}^*(\mathbf{X}))(\langle t, s \rangle)$ is defined as the image ring $\mathbf{Im}(\mathbf{H}^*(\mathbf{X})([t - \delta, s + \delta])) = \mathbf{Im}(\mathbf{H}^*(\mathbb{X}_{s+\delta}) \rightarrow \mathbf{H}^*(\mathbb{X}_{t-\delta}))$ for sufficiently small $\delta > 0$, when $\langle t, s \rangle = (t, s)$, and is defined similarly for the cases when $\langle t, s \rangle = (t, s]$ or $[t, s)$.

► **Remark 9.** It follows from [11, Prop. 38] that the cup-length invariant **cup** is non-increasing under surjective morphisms and non-decreasing under injective morphisms, which we call an *inj-surj invariant*. As a consequence, for any persistent space \mathbf{X} , the persistent cup-length function $\text{cup}(\mathbf{X})$ defines a *functor* from (\mathbf{Int}, \leq) to (\mathbb{N}, \geq) .

Prop. 10 below allows us to compute the cohomology images of a persistent cohomology ring from representative cocycles, which will be applied to compute persistent cup-length functions in Ex. 12 and prove Thm. 1 in the full version of this paper, see [11, page 24]. Prop. 11 allows us to simplify the calculation of persistent cup-length functions in certain cases, such as the Vietoris-Rips filtration of products or wedge sums of metric spaces.

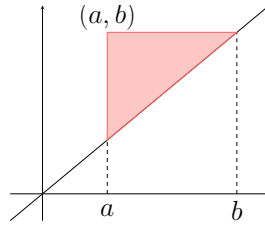
► **Proposition 10** (Persistent image ring). *Let $\mathbf{X} = \{\mathbb{X}_t\}_{t \in \mathbb{R}}$ be a filtration, together with a family of representative cocycles $\sigma = \{\sigma_I\}_{I \in \mathcal{B}(\mathbf{X})}$ for $\mathbf{H}^*(\mathbf{X})$. Let $t \leq s$ in \mathbb{R} . Then $\mathbf{Im}(\mathbf{H}^*(\mathbb{X}_s) \rightarrow \mathbf{H}^*(\mathbb{X}_t)) = \langle [\sigma_I]_t : [t, s] \subset I \in \mathcal{B}(\mathbf{X}) \rangle$, generated as a graded ring.*

► **Proposition 11.** *Let $\mathbf{X}, \mathbf{Y} : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ be two persistent spaces. Then:*

- $\mathbf{cup}(\mathbf{X} \times \mathbf{Y}) = \mathbf{cup}(\mathbf{X}) + \mathbf{cup}(\mathbf{Y})$,
- $\mathbf{cup}(\mathbf{X} \amalg \mathbf{Y}) = \max\{\mathbf{cup}(\mathbf{X}), \mathbf{cup}(\mathbf{Y})\}$, and
- $\mathbf{cup}(\mathbf{X} \vee \mathbf{Y}) = \max\{\mathbf{cup}(\mathbf{X}), \mathbf{cup}(\mathbf{Y})\}$.

Here \times, \amalg and \vee denote point-wise product, disjoint union, and wedge sum, respectively.

Examples and visualization. Each interval $\langle a, b \rangle$ in \mathbf{Int} is visualized as a point (a, b) in the half-plane above the diagonal (see Fig. 3). To visualize the persistent cup-length function of a filtration \mathbf{X} , we assign to each point (a, b) the integer value $\mathbf{cup}(\mathbf{X})(\langle a, b \rangle)$, if it is positive. If $\mathbf{cup}(\mathbf{X})(\langle a, b \rangle) = 0$ we do not assign any value. We present an example to demonstrate how persistent cup-length functions are visualized in the upper-diagonal plane (see Fig. 1).



■ **Figure 3** The interval $\langle a, b \rangle$ in \mathbf{Int} corresponds to the point (a, b) in \mathbb{R}^2 .

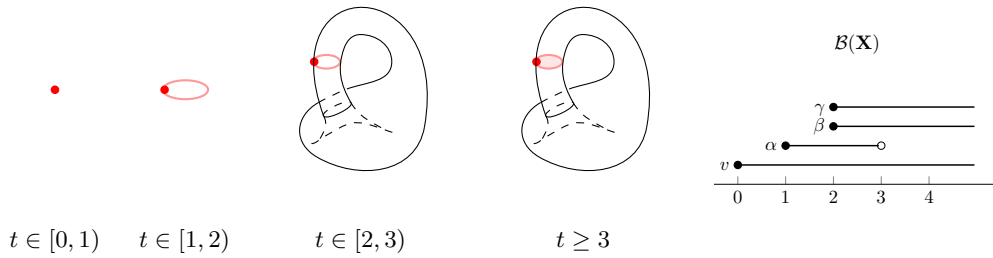
► **Example 12** (Visualization of $\mathbf{cup}(\cdot)$). Recall the filtration $\mathbf{X} = \{\mathbb{X}_t\}_{t \geq 0}$ of a Klein bottle with a 2-cell attached, defined in Fig. 1. Consider the persistent cohomology $\mathbf{H}^*(\mathbf{X})$ in \mathbb{Z}_2 -coefficients. Let v be the 0-cocycle born at $t = 0$, let α be the 1-cocycle born at $t = 1$ and died at $t = 3$, and let β be the 1-cocycle born at time $t = 2$. Let $\gamma := \beta \smile \beta$, which is then a non-trivial 2-cocycle born at time $t = 2$, like β . Then the barcodes of \mathbf{X} are: $\mathcal{B}_0(\mathbf{X}) = \{[0, \infty)\}$, $\mathcal{B}_1(\mathbf{X}) = \{[1, 3), [2, \infty)\}$, and $\mathcal{B}_2(\mathbf{X}) = \{[2, \infty)\}$. See Fig. 4.

Using the formula in Prop. 10, for any $t \leq s$, we have

$$\mathbf{Im}(\mathbf{H}^*(\mathbb{X}_s) \rightarrow \mathbf{H}^*(\mathbb{X}_t)) = \begin{cases} \langle [v]_t, [\beta]_t, [\gamma]_t \rangle, & \text{if } 2 \leq t < 3 \text{ and } s \geq 3 \\ \langle [v]_t, [\alpha]_t, [\beta]_t, [\gamma]_t \rangle, & \text{if } 2 \leq t \leq s < 3 \\ \langle [v]_t, [\alpha]_t \rangle, & \text{if } 1 \leq t < 2 \text{ and } s < 3 \\ \langle [v]_t \rangle, & \text{otherwise.} \end{cases}$$

The persistent cup-length function of \mathbf{X} is computed as follows and visualized in Fig. 1.

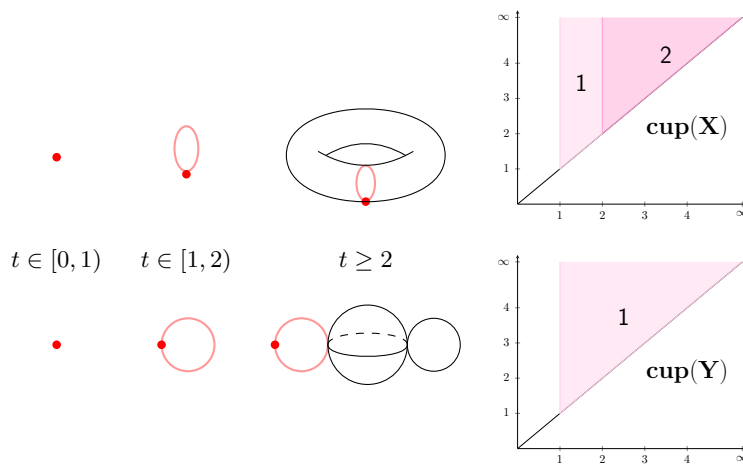
$$\mathbf{cup}(\mathbf{X})([t, s]) = \begin{cases} 2, & \text{if } t \geq 2 \\ 1, & \text{if } 1 \leq t < 2 \text{ and } s < 3 \\ 0, & \text{otherwise.} \end{cases}$$



■ **Figure 4** The filtration \mathbf{X} given in Fig. 1 and its barcode $\mathcal{B}(\mathbf{X})$, see Ex. 12.

We end this section by presenting an example, Ex. 13, where the persistent cup-length function distinguishes a pair of filtrations which the total barcode is not able to. A similar example is available in [11, §D.2], where we will also give a quantitative measure via the erosion distance on the difference between persistent cup-length functions of different filtrations.

► **Example 13 (cup(\cdot) better than standard barcode).** Consider the filtration $\mathbf{X} = \{\mathbb{X}_t\}_{t \geq 0}$ of a 2-torus \mathbb{T}^2 and the filtration $\mathbf{Y} = \{\mathbb{Y}_t\}_{t \geq 0}$ of the space $\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1$ as shown in Fig. 5. Knowing that $\mathbb{X}_3 = \mathbb{T}^2$ and $\mathbb{Y}_3 = \mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1$ have the same (co)homology vector spaces in all dimensions, one can directly check that the persistent (co)homology vector spaces associated to \mathbf{X} and \mathbf{Y} are the same. However, the cohomology ring structure of \mathbb{X}_3 is different from that of \mathbb{Y}_3 : there are two 1-cocycles in \mathbb{X}_3 with a non-zero product (indeed the product is a 2-cocycle), whereas all 1-cocycles in \mathbb{Y}_3 have zero product. This difference between the cohomology ring structures of these two filtration is quantified by their persistent cup-length functions, see Fig. 5. Also, see [11, Ex. 54] for a more geometric example, which considers the Vietoris-Rips filtrations of \mathbb{T}^2 and $\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1$.



■ **Figure 5** Top: A filtration \mathbf{X} of \mathbb{T}^2 and its persistent cup-length function $\text{cup}(\mathbf{X})$. Bottom: A filtration \mathbf{Y} of the wedge sum $\mathbb{S}^1 \vee \mathbb{S}^2 \vee \mathbb{S}^1$ and its persistent cup-length function $\text{cup}(\mathbf{Y})$. See Ex. 13.

3 The persistent cup-length diagram

In this section, we introduce the notion of the *persistent cup-length diagram of a filtration*, by using the cup product operation on cocycles. In §3.1, we show how the persistent cup-length diagram is used to compute the persistent cup-length function (see Thm. 1). In §3.2, we develop an algorithm (see Alg. 3) to compute the persistent cup-length diagram, and study its complexity. Proofs, details and extra examples are available in [11, §C].

3.1 Persistent cup-length diagram

We define the persistent cup-length diagram using a family of representative cocycles. In Thm. 1 we show that the persistent cup-length function can be retrieved from the persistent cup-length diagram. See [11, §C.1] for the proofs of Thm. 1 and details for this section.

► **Definition 14** (ℓ -fold $*_{\sigma}$ -product). *Let σ be a family of representative cocycles for $\mathbf{H}^*(\mathbf{X})$. Let $\ell \in \mathbb{N}^*$ and let I_1, \dots, I_ℓ be a sequence of elements in $\mathcal{B}(\mathbf{X})$ with representative cocycles $\sigma_{I_1}, \dots, \sigma_{I_\ell} \in \sigma$, respectively. We define the ℓ -fold $*_{\sigma}$ -product of I_1, \dots, I_ℓ to be*

$$I_1 *_{\sigma} \cdots *_{\sigma} I_\ell := \{t \in \mathbb{R} \mid [\sigma_{I_1}]_t \smile \cdots \smile [\sigma_{I_\ell}]_t \neq [0]_t\}, \quad (4)$$

associated with the formal representative cocycle $\sigma_{I_1} \smile \cdots \smile \sigma_{I_\ell}$. We also call the right-hand side of Eq. (4) the **support** of $\sigma_{I_1} \smile \cdots \smile \sigma_{I_\ell}$, and denote it by $\text{supp}(\sigma_{I_1} \smile \cdots \smile \sigma_{I_\ell})$.

The support of a product of representative cocycles is always an interval:

► **Proposition 15** (Support is an interval). *With the same assumption and notation in Defn. 14, let $I := \text{supp}(\sigma_{I_1} \smile \cdots \smile \sigma_{I_\ell})$. If $I \neq \emptyset$, then I is an interval $\langle b, d \rangle$, where $b \leq d$ are such that d is the right end of $\bigcap_{1 \leq i \leq \ell} I_i$ and b is the left end of some $J \in \mathcal{B}(\mathbf{X})$ (J is not necessarily one of the I_i).*

The $*_{\sigma}$ -product is associative and invariant under permutations. Let $\mathcal{B}_{\geq 1}(\mathbf{X}) := \sqcup_{p \geq 1} \mathcal{B}_p(\mathbf{X})$. Let $\mathcal{B}_{\geq 1}(\mathbf{X})^{*\sigma^\ell}$ be the set of $I_1 *_{\sigma} \cdots *_{\sigma} I_\ell$ where each $I_i \in \mathcal{B}_{\geq 1}(\mathbf{X})$. For the simplicity of notation, we often write $\mathcal{B}_{\geq 1}(\mathbf{X})^{*\sigma^\ell}$ as $\mathcal{B}(\mathbf{X})^{*\sigma^\ell}$.

► **Definition 16** (persistent cup-length diagram). *Let \mathbf{X} be a filtration and let $\mathcal{B}_{\geq 1}(\mathbf{X})$ be its barcode over positive dimensions. Let $\sigma = \{\sigma_I\}_{I \in \mathcal{B}_{\geq 1}(\mathbf{X})}$ be a family of representative cocycles for $\mathbf{H}^{\geq 1}(\mathbf{X})$. The **persistent cup-length diagram of \mathbf{X} (associated to σ)** is defined to be the map $\mathbf{dgm}_{\sigma}^{\smile}(\mathbf{X}) : \text{Int} \rightarrow \mathbb{N}$, given by:*

$$\mathbf{dgm}_{\sigma}^{\smile}(\mathbf{X})(I) := \max\{\ell \in \mathbb{N}^* \mid I = I_1 *_{\sigma} \cdots *_{\sigma} I_\ell, \text{ where each } I_i \in \mathcal{B}_{\geq 1}(\mathbf{X})\},$$

with the convention that $\max \emptyset = 0$.

Recall Thm. 1, which states the relation between the persistent cup-length function $\mathbf{cup}(X)$ and the persistent cup-length diagram $\mathbf{dgm}_{\sigma}^{\smile}(\mathbf{X})$: for any interval $\langle a, b \rangle$, the $\mathbf{cup}(\mathbf{X})(\langle a, b \rangle)$ attains the **maximum** value of $\mathbf{dgm}_{\sigma}^{\smile}(\mathbf{X})(\langle c, d \rangle)$ over all intervals $\langle c, d \rangle \supseteq \langle a, b \rangle$. This is in the same spirit as in [32] where the rank function can be reconstructed from the persistence diagram by replacing “max” operation with the **sum** operation.

► **Example 17** (Example of $\mathbf{dgm}_{\sigma}^{\smile}(\cdot)$ and Thm. 1). Recall the filtration $\mathbf{X} = \{\mathbb{X}_t\}_{t \geq 0}$ of the pinched Klein bottle defined in Fig. 1, and its persistent cup-length function and the representative cocycles $\{\alpha, \beta, \gamma\} =: \sigma$ from Ex. 12. Because $\mathbf{H}^*(\mathbf{X})$ is non-trivial up to dimension 2, $\mathbf{dgm}_{\sigma}^{\smile}(\mathbf{X})(I) \leq 2$ for any I . It follows from $[\alpha \smile \alpha] = 0$ that

$\text{dgm}_\sigma^\smile(\mathbf{X})([1, 3]) = 1$, and from $[\alpha \smile \beta] = [\gamma]$ that $[2, 3] = [1, 3] *_\sigma [2, \infty)$, implying $\text{dgm}_\sigma^\smile(\mathbf{X})([2, 3]) = 2$. A similar argument holds for $[2, \infty)$, using the fact that $[\beta \smile \beta] = [\gamma]$. Thus, we obtain the persistent cup-length diagram $\text{dgm}_\sigma^\smile(\mathbf{X})$ as below (see the right-most figure in Fig. 1 for its visualization):

$$\text{dgm}_\sigma^\smile(\mathbf{X})(I) = \begin{cases} 1, & \text{if } I = [1, 3) \\ 2, & \text{if } I = [2, 3) \text{ or } I = [2, \infty) \\ 0, & \text{otherwise.} \end{cases}$$

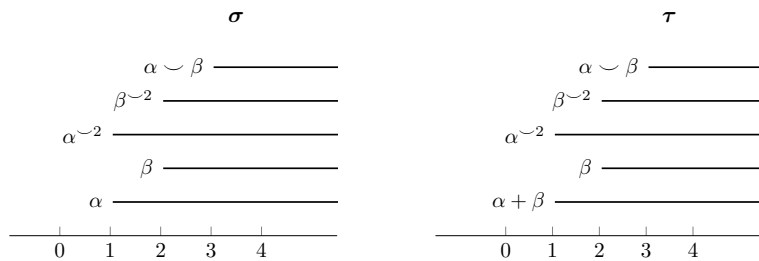
Applying Thm. 1, we obtain the persistent cup-length function $\text{cup}(\mathbf{X})$ shown in the middle figure of Fig. 1.

See [11, §C.1] for the proof of Thm. 1 and more examples of persistent cup-length diagrams. It is worth noticing that the persistent cup-length diagram depends on the choice of the family of representative cocycles σ , see Ex. 18 below.

► **Example 18** ($\text{dgm}_\sigma^\smile(\mathbf{X})$ depends on σ). Let $\mathbb{R}\mathbb{P}^2$ be the real projective plane. Consider the filtration \mathbf{X} of the 2-skeleton $S_2(\mathbb{R}\mathbb{P}^2 \times \mathbb{R}\mathbb{P}^2)$ of the product space $\mathbb{R}\mathbb{P}^2 \times \mathbb{R}\mathbb{P}^2$, given by:

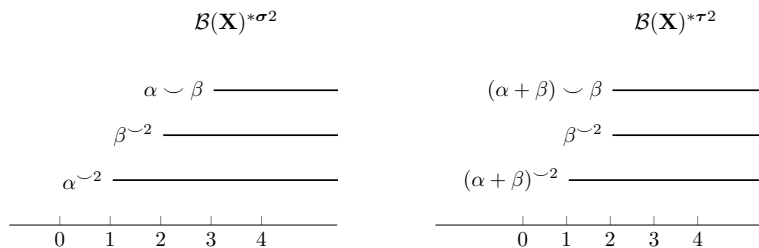
$$\mathbf{X} : \quad \bullet \xrightarrow{t \in [0, 1)} \mathbb{R}\mathbb{P}^2 \xrightarrow{t \in [1, 2)} \mathbb{R}\mathbb{P}^2 \vee \mathbb{R}\mathbb{P}^2 \xrightarrow{t \in [2, 3)} S_2(\mathbb{R}\mathbb{P}^2 \times \mathbb{R}\mathbb{P}^2) \xrightarrow{t \geq 3}$$

Let α be the 1-cocycle born at $t = 1$, and β be the 1-cocycles born at $t = 2$ when the second copy of $\mathbb{R}\mathbb{P}^2$ appears. See Fig. 6 for two choices of representative cocycles σ and τ for $\mathcal{B}_{\geq 1}(\mathbf{X})$, where these two choices only differ by the first dimensional cocycles associated with the bar $[1, \infty)$. For a detailed explanation of the cohomology rings of the above spaces, see [11, §C.1].



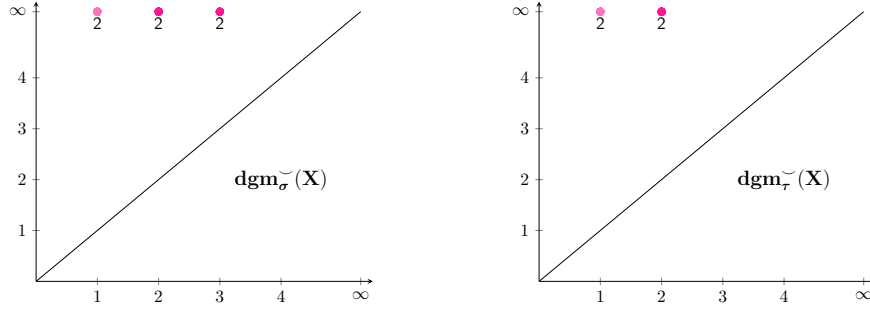
■ **Figure 6** Two choices of representative cocycles for the filtration \mathbf{X} given by Ex. 18.

To obtain the cup-length diagram, we first compute $\mathcal{B}(\mathbf{X})^{*\sigma^2}$ and $\mathcal{B}(\mathbf{X})^{*\tau^2}$:



By Defn. 16, the persistent cup-length diagram associated to σ and τ are (see Fig. 7):

31:12 Persistent Cup-Length



■ **Figure 7** The persistent cup-length diagrams $\mathbf{dgm}_{\sigma}^{\sim}(\mathbf{X})$ (left) and $\mathbf{dgm}_{\tau}^{\sim}(\mathbf{X})$ (right), see Ex. 18.

See [11, §C.2] for more examples of persistent cup-length diagrams. In the next section, we develop an algorithm for computing the persistent cup-length diagram $\mathbf{dgm}_{\sigma}^{\sim}(\mathbf{X})$, which can be used to compute the persistent cup-length function $\mathbf{cup}(\mathbf{X})$ due to Thm. 1.

3.2 An algorithm for computing the persistent cup-length diagram over \mathbb{Z}_2

Let $\mathbf{X} : \mathbb{X}_1 \hookrightarrow \cdots \hookrightarrow \mathbb{X}_N (= \mathbb{X})$ be a finite filtration of a finite simplicial complex \mathbb{X} . Suppose that the barcode over positive dimensions $\mathcal{B} := \mathcal{B}_{\geq 1}(\mathbf{X})$ and a family of representative cocycles $\sigma := \{\sigma_I\}_{I \in \mathcal{B}}$ are given. Because a finite filtration has only finitely many critical values, we assume that all intervals in the barcode are closed at the right end. If not, we replace the right end of each such interval with its closest critical value to the left. Since each interval is considered together with a representative cocycle in this section, we will abuse the notation and write \mathcal{B} for the set $\{(I, \sigma_I)\}_{I \in \mathcal{B}}$ as well. Let $\mathcal{B}^{*\sigma^\ell}$ be the set of $I_1 *_{\sigma} \cdots *_{\sigma} I_\ell$ where each $I_i \in \mathcal{B}$. We compute $\{\mathcal{B}^{*\sigma^\ell}\}_{\ell \geq 1}$ using:

■ **Algorithm 2** Computing $\{\mathcal{B}^{*\sigma^\ell}\}_{\ell \geq 1}$.

```

1 while  $\mathcal{B}^{*\sigma^\ell} \neq \emptyset$  do
2   for  $(I_1, \sigma_1) \in \mathcal{B}$  and  $(I_2, \sigma_2) \in \mathcal{B}^{*\sigma^\ell}$  do
3     if  $I_1 *_{\sigma} I_2 \neq \emptyset$  then
4       Append  $(I_1 *_{\sigma} I_2, \sigma_1 \smile \sigma_2)$  to  $\mathcal{B}^{*\sigma^{(\ell+1)}}$ 

```

The **line 3** involves the computation of $\text{supp}(\sigma_1 \smile \sigma_2)$ which is some interval $[b_\sigma, d_\sigma]$ for $1 \leq b_\sigma \leq d_\sigma \leq m$ such that d_σ is simply the right end of $I_1 \cap I_2$ and b_σ is the left end of some $I \in \mathcal{B}$, by Prop. 15. The computation of b_σ is broken down in two steps: (1) compute the cup product (at cochain level) $\sigma := \sigma_1 \smile \sigma_2$, and (2) find b_σ as the smallest $i \leq d_\sigma$ such that $\sigma|_{C_*(\mathbb{X}_i)}$ is not a coboundary. Step (1) is already addressed by Alg. 1 on page 6. Let us now introduce an algorithm to address Step (2).

3.3 Checking whether a cochain is a coboundary

As before, we assume a total order (e.g. the order given in [5]) on the simplices, and denote the ordered simplices by $S = \{\alpha_1 < \cdots < \alpha_m\}$, where m is the number of simplices. We adopt the reverse ordering for the set of cosimplices $S^* := \{\alpha_m^* < \cdots < \alpha_1^*\}$. Notice that S^* forms a basis for the linear space of cochains. A p -cochain σ is written as a linear sum of elements in S^* uniquely. If α_j^* appears as a summand for σ , we denote $\alpha_j \in \sigma$.

Let A be the coboundary matrix associated to the ordered basis S^* . Assume that $R = AV$ is the reduced matrix of A obtained from left-to-right column operations, given by the upper triangular matrix V . As a consequence, the pivots $\text{Pivots}(R)$ of columns of R are unique. Using all i -th row for $i \in \text{Pivots}(R)$, we do bottom-to-top row reduction on R : $UR = \Lambda$, such that Λ has at most one non-zero element in each row and column, and U is an upper triangular matrix. See [11, Alg. 3] for the row reduction algorithm (with complexity $O(m^2)$), which outputs the matrix U . The following proposition allows us to use the row reduction matrix U and $\text{Pivots}(R)$ to check whether a cochain is a coboundary. For the proof of Prop. 19, see [11, §C.3].

► **Proposition 19.** *Given a p -cochain σ , let $y \in \mathbb{Z}_2^m$ be such that $\sigma = S^* \cdot y$. Let R be the column reduced coboundary matrix, U be the row reduction matrix of R . Then σ is a coboundary, iff $\{i : \text{the } i\text{-th row of } (U \cdot y) \neq 0\} \subset \text{Pivots}(R)$.*

Using the boundary matrix. We can also use the boundary matrix to check whether a cocycle is a coboundary, see [11, page 28]. Using the boundary matrix, only column reduction is needed, while for the coboundary matrix both column reduction and row reduction are performed. However, it has been justified in [5] that reducing the coboundary matrix is more efficient than reducing the boundary matrix. Combined with the fact that the row reduction step does not increase the computation complexity of computing the persistent cup-length diagram, we will use the coboundary matrix in this paper.

3.4 Main algorithm and its complexity

Motivated by the goal of obtaining a practical algorithm, and in order to control the complexity, we consider truncating filtrations up to a user specified dimension bound.

Truncation of a filtration. Fix a positive integer k . Given a filtration $\mathbf{X} : \mathbb{X}_1 \hookrightarrow \dots \hookrightarrow \mathbb{X}_N (= \mathbb{X})$ of a finite simplicial complex \mathbb{X} , let \mathbb{X}_i^{k+1} be the $(k+1)$ -skeleton of \mathbb{X}_i for each i . The $(k+1)$ -dimensional truncation of \mathbf{X} is the filtration $\mathbf{X}^{k+1} : \mathbb{X}_1^{k+1} \hookrightarrow \dots \hookrightarrow \mathbb{X}_N^{k+1}$. Since $\mathbf{H}^{\leq k}(\mathbb{Y}) \cong \mathbf{H}^{\leq k}(\mathbb{Y}^{k+1})$ (as vector spaces) for any simplicial complex \mathbb{Y} , we conclude that $\mathbf{H}^{\leq k}(\mathbf{X}) \cong \mathbf{H}^{\leq k}(\mathbf{X}^{k+1})$ as persistent vector spaces. Thus, the barcode $\mathcal{B}(\mathbf{X}^{k+1})$ of \mathbf{X}^{k+1} is equal to the barcode $\mathcal{B}_{\leq k}(\mathbf{X}) := \sqcup_{p \leq k} \mathcal{B}_p(\mathbf{X})$. Let $\mathcal{B}_{[1,k]}(\mathbf{X}) := \sqcup_{1 \leq p \leq k} \mathcal{B}_p(\mathbf{X})$. We introduce Alg. 3 for computing the persistent cup-length diagram for the $(k+1)$ -dimensional truncation of \mathbf{X} over \mathbb{Z}_2 . The time complexity of Alg. 3 is described in terms of the variables below:

- k is a dimension bound used to truncate the filtration;
- m_k is the number of simplices with positive dimension in the $(k+1)$ -skeleton \mathbb{X}^{k+1} of \mathbb{X} ;
- c_k is the complexity of checking whether a simplex is alive at a given filtration parameter;
- $q_{k-1} := \max_{1 \leq \ell \leq k-1} \text{card} \left((\mathcal{B}_{[1,k]}(\mathbf{X}))^{*\sigma^\ell} \right)$ (see Defn. 14). In particular, $q_1 = \text{card}(\mathcal{B}_{[1,k]}(\mathbf{X}))$.

Time complexity. In Alg. 3, line 9 runs no more than $q_1 \cdot q_{k-1}$ times, due to the definition of q_1 and q_{k-1} . The while loop in line 14 runs no more than $\text{card}(\text{b_time}) \leq q_1$ times, and the condition of this while loop involves a matrix multiplication whose complexity is at most $O((m_k)^2)$. Combined with other comments in Alg. 3 and the fact that k is a fixed constant, the total complexity is upper bounded by

$$O(k) \cdot O(q_1 \cdot q_{k-1}) \cdot O((m_k)^2 \cdot \max\{c_k, q_1\}) \leq O((m_k)^2 \cdot q_1 \cdot q_{k-1} \cdot \max\{c_k, q_1\}).$$

■ **Algorithm 3** Main algorithm: compute persistent cup-length diagram.

Input : A dimension bound k , the ordered list of cosimplices S^* from dimension 1 to $k + 1$, the column reduced coboundary matrix R from dimension 1 to $k + 1$, and barcodes (annotated by representative cocycles) from dimension 1 to k : $\mathcal{B}_{[1,k]} = \{(b_\sigma, d_\sigma, \sigma)\}_{\sigma \in \sigma}$, where each σ is a representative cocycle for the bar (b_σ, d_σ) and $\{\sigma_1, \dots, \sigma_{q_1}\}$ is ordered first in the increasing order of the death time and then in the increasing order of the birth time.

Output : A matrix representation A_ℓ of persistent cup-length diagram, and the lists of distinct birth times `b_time` and death times `d_time`.

```

1 b_time, d_time ← unique( $\{b_\sigma\}_{\sigma \in \sigma}$ ), unique( $\{d_\sigma\}_{\sigma \in \sigma}$ );
2  $m_k, \ell, B_1 \leftarrow \text{card}(S^*), 1, \mathcal{B}_{[1,k]}$ ;
3  $A_0 = A_1 \leftarrow \text{zeros}(\text{card}(\text{b\_time}), \text{card}(\text{d\_time}))$ ;
4  $U \leftarrow \text{RowReduce}(R)$ ; //  $O(m_k^2)$ , [11, Alg. 3]
5 for  $(b_i, b_j) \in B_1$  do //  $O(m_k)$ 
6    $A_1(i, j) \leftarrow 1$ ;
7 while  $A_{\ell-1} \neq A_\ell$  and  $\ell \leq k - 1$  do //  $O(k)$ 
8    $B_{\ell+1} = \{\}$ ;
9   for  $(b_{i_1}, d_{j_1}, \sigma_1) \in B_1$  and  $(b_{i_2}, d_{j_2}, \sigma_2) \in B_\ell$  do //  $O(q_1 \cdot q_{k-1})$ 
10     $\sigma \leftarrow \text{CupProduct}(\sigma_1, \sigma_2, S^*)$ ; //  $O(m_k^2 \cdot c_k)$ , Alg. 1
11     $y \leftarrow$  the vector representation of  $\sigma$  in  $S^*$ ;
12     $i \leftarrow \max\{i' : b_{i'} \leq d_{\min\{j_1, j_2\}}\}$ ;
13     $s_i \leftarrow$  number of simplices alive at  $b_i$ ;
14    while  $\{u : (U_{m_k+1-s_i:m_k, m_k+1-s_i:m_k} \cdot y_{m_k+1-s_i:m_k})(u) \neq 0\} \subset \text{Pivots}(R)$  //  $O((m_k)^2 \cdot q_1)$ 
15     do
16      $i \leftarrow i - 1$ ;
17      $s_i \leftarrow$  number of simplices alive at  $b_i$ ;
18     if  $b_i < d_{\min\{j_1, j_2\}}$  then
19     Append  $(b_i, d_{\min\{j_1, j_2\}}, \sigma)$  to  $B_{\ell+1}$ ;
20      $A_{\ell+1}(i, \min\{j_1, j_2\}) \leftarrow \ell$ 
21    $\ell \leftarrow \ell + 1$ .
22 return  $A_\ell, \text{b\_time}, \text{d\_time}$ .
```

Next, we estimate q_{k-1} and c_k using q_1 , m_k and k . Since each B_ℓ consists of ℓ -fold \ast_σ -products of elements in B_1 , we have $q_{k-1} = \max_{1 \leq \ell \leq k-1} \text{card}(B_\ell) \leq (q_1)^{k-1}$, which turns out to be a very coarse bound (see [11, Rmk. 45]). On the other hand, c_k as the cost of checking whether a simplex is alive at a given filtration, is at most m_k the number of simplices. Hence, the complexity of Alg. 3 is upper bounded by $O((m_k)^3 \cdot q_1^k)$. In addition, we have $q_1 \leq m_k$, because in the matrix reduction algorithm for computing barcodes, bars are obtained from the pivots of the column reduced coboundary matrix and each column provides at most one pivot. Thus, $O((m_k)^2 \cdot q_1 \cdot q_{k-1} \cdot \max\{c_k, q_1\}) \leq O((m_k)^3 \cdot q_1^k) \leq O((m_k)^{k+3})$.

Consider the Vietoris-Rips filtration arising from a metric space of n points with the distance matrix D . Then **line 7** of Alg. 1, checking whether a simplex a (represented by a set of at most $k + 1$ indices into $[n]$) is alive at the filtration parameter value t , can be done by checking whether $\max(D[a, a]) \leq t$, with constant time complexity $c_k = O(k^2)$. In summary, we have the following theorem.

► **Theorem 20** (Complexity of Alg. 3). *For an arbitrary finite filtration truncated up to dimension $(k + 1)$, computing its persistent cup-length diagram via Alg. 3 has complexity at most $O((m_k)^2 \cdot q_1 \cdot q_{k-1} \cdot \max\{c_k, q_1\})$. In terms of just m_k , the complexity of Alg. 3 is at most $O((m_k)^{k+3})$, since $c_k \leq m_k$ and $q_{k-1} \leq (m_k)^{k-1}$.*

For the $(k + 1)$ -dimensional truncation of the Vietoris-Rips filtration arising from a metric space of n points, the complexity of Alg. 3 is improved to $O((m_k)^2 \cdot q_1^2 \cdot q_{k-1})$, which is at most $O((m_k)^{k+3}) \leq O(n^{k^2+5k+6})$.

Notice that when $k = 1$, the persistent cup-length diagram simply evaluates 1 at each bar in the standard barcode, and 0 elsewhere. When $k \geq 2$, the resulting persistent cup-length diagram becomes more informative and captures certain topological features that the standard persistence diagram is not able to detect. This is reflected in [11, Ex. 42].

Although the algorithm has not been tested on datasets yet, it is a practical algorithm, given that there are available software programs, such as Ripser (see [4]), which compute the barcode and extracts representative cocycles for Vietoris-Rips filtrations. Note that, according to [5], the implementation ideas ‘are also applicable to persistence computations for other filtrations as well’.

► **Remark 21** (Estimating the parameter q_{k-1}). The inequality $q_{k-1} \leq (m_k)^{k-1}$ is quite coarse in general. Consider a filtration consisting of contractible spaces, where q_{k-1} is always 0 but m_k can be arbitrarily large. Even in the case when there is a reasonable number of cohomology classes with non-trivial cup products, q_{k-1} can be much smaller than $(m_k)^{k-1}$. See [11, Rmk. 45].

► **Remark 22** (Reducing the time complexity). Because cup products cannot live longer than their factors, discarding short bars will not result into loss of important information. In our algorithm, an extra parameter $\epsilon \geq 0$ can be added to discard all the bars in the barcode B_1 with length less than ϵ . By doing so, since the cardinality of B_1 is decreased, one expects the runtime of Alg. 3 (in particular inside the loop in **line 9**) to be significantly reduced. A similar trimming strategy can also be applied in the construction of the subsequent B_ℓ s.

Correctness of the algorithm. Checking whether a cocycle is a coboundary requires local matrix reduction for the given filtration parameter $d_{\min\{j_1, j_2\}}$, but a global matrix reduction is performed in the algorithm. The reason is that the coboundary matrix A , the column reduction matrix V and the row reduction matrix U are all upper-diagonal. Therefore, reducing the ambient matrix A and then taking the bottom-right submatrix to get \bar{U} , is equivalent to reducing the submatrix of A directly.

References

- 1 Henry Adams, Andrew Tausz, and Mikael Vejdemo-Johansson. javaplex: A research software package for persistent (co)homology. In *Mathematical Software, ICMS 2014 - 4th International Congress, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 129–136. Springer Verlag, 2014. 4th International Congress on Mathematical Software, ICMS 2014 ; Conference date: 05-08-2014 Through 09-08-2014. doi:10.1007/978-3-662-44199-2_23.
- 2 Manuel Amann. Computational complexity of topological invariants. *Proceedings of the Edinburgh Mathematical Society*, 58(1):27–32, 2015. doi:10.1017/S0013091514000455.
- 3 HB Aubrey. *Persistent cohomology operations*. PhD thesis, Duke University, 2011.
- 4 Ulrich Bauer. Ripser. <https://github.com/Ripser/ripser>, 2016.
- 5 Ulrich Bauer. Ripser: efficient computation of Vietoris–Rips persistence barcodes. *Journal of Applied and Computational Topology*, pages 1–33, 2021. doi:10.1007/s41468-021-00071-5.

- 6 Francisco Belchí and Anastasios Stefanou. A-infinity persistent homology estimates detailed topology from point cloud datasets. *Discrete & Computational Geometry*, pages 1–24, 2021. doi:10.1007/s00454-021-00319-y.
- 7 Andrew J Blumberg and Michael Lesnick. Universality of the homotopy interleaving distance. *arXiv preprint*, 2017. arXiv:1705.01690.
- 8 Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.
- 9 Gunnar Carlsson. Persistent homology and applied homotopy theory. In *Handbook of Homotopy Theory*, pages 297–329. Chapman and Hall/CRC, 2020.
- 10 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007. doi:10.1007/s00454-006-1276-5.
- 11 Marco Contessoto, Facundo Mémoli, Anastasios Stefanou, and Ling Zhou. Persistent cup-length. *arXiv preprint*, 2021. arXiv:2107.01553.
- 12 Luis Polanco Contreras and Jose Perea. Persistent cup product for quasi periodicity detection. https://4c0aa4c9-c4b2-450c-a81a-c4a8e2d3f528.filesusr.com/ugd/58704f_dcd2001732bb4b3ab91900f99955241c.pdf, 2021. Second Graduate Student Conference: Geometry and Topology meet Data Analysis and Machine Learning (GTDAML2021).
- 13 Octavian Cornea, Gregory Lupton, John Oprea, Daniel Tanré, et al. *Lusternik-Schnirelmann category*. Number 103 in Mathematical Surveys and Monographs. American Mathematical Society, 2003.
- 14 William Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules. *Journal of Algebra and its Applications*, 14(05):1550066, 2015. doi:10.1142/S0219498815500668.
- 15 Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27(12):124003, November 2011. doi:10.1088/0266-5611/27/12/124003.
- 16 Vin De Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Discrete & Computational Geometry*, 45(4):737–759, 2011. doi:10.1007/s00454-011-9344-x.
- 17 Paweł Dłotko and Hubert Wagner. Simplification of complexes for persistent homology computations. *Homology, Homotopy and Applications*, 16(1):49–63, 2014. doi:10.4310/HHA.2014.v16.n1.a3.
- 18 Herbert Edelsbrunner and John Harer. Persistent homology—a survey. *Contemporary mathematics*, 453:257–282, 2008.
- 19 Michael Farber. Topological complexity of motion planning. *Discrete and Computational Geometry*, 29(2):211–221, 2003. doi:10.1007/s00454-002-0760-9.
- 20 Patrizio Frosini. A distance for similarity classes of submanifolds of a euclidean space. *Bulletin of the Australian Mathematical Society*, 42(3):407–415, 1990. doi:10.1017/S0004972700028574.
- 21 Patrizio Frosini. Measuring shapes by size functions. In *Intelligent Robots and Computer Vision X: Algorithms and Techniques*, volume 1607, pages 122–133. International Society for Optics and Photonics, 1992. doi:10.1117/12.57059.
- 22 Rocío González Díaz and Pedro Real Jurado. Computation of cohomology operations of finite simplicial complexes. *Homology, Homotopy and Applications (HHA)*, 5 (2), 83–93., 2003.
- 23 Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000. URL: <https://cds.cern.ch/record/478079>.
- 24 Estanislao Herscovich. A higher homotopic extension of persistent (co)homology. *Journal of Homotopy and Related Structures*, 13(3):599–633, 2018. doi:10.1007/s40062-017-0195-x.
- 25 Jonathan Huang. Cup products in computational topology, 2005. URL: <http://jonathan-huang.org/research/old/computationalcupproduct.pdf>.

- 26 Tomasz Kaczynski, Paweł Dłotko, and Marian Mrozek. Computing the cubical cohomology ring. *Image-A: Applicable Mathematics in Image Engineering*, 1 (3), 137–142, 2010. URL: <http://hdl.handle.net/11441/26211>.
- 27 Louis Kang, Boyan Xu, and Dmitriy Morozov. Evaluating state space discovery by persistent cohomology in the spatial representation system. *Frontiers in Computational Neuroscience*, 15, 2021. doi:10.3389/fncom.2021.616748.
- 28 Woojin Kim and Facundo Mémoli. Spatiotemporal persistent homology for dynamic metric spaces. *Discrete & Computational Geometry*, 66(3):831–875, 2021. doi:10.1007/s00454-019-00168-w.
- 29 Umberto Lupo, Anibal M. Medina-Mardones, and Guillaume Tauzin. Persistence Steenrod modules. *arXiv preprint*, pages arXiv–1812, 2018. arXiv:1812.05031.
- 30 Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The Gudhi library: Simplicial complexes and persistent homology. In Hoon Hong and Chee Yap, editors, *Mathematical Software – ICMS 2014*, pages 167–174, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. doi:10.1007/978-3-662-44199-2_28.
- 31 James R. Munkres. *Elements of algebraic topology*. Addison-Wesley, Menlo Park, CA, 1984. doi:10.1201/9780429493911.
- 32 Amit Patel. Generalized persistence diagrams. *Journal of Applied and Computational Topology*, 1(3):397–419, 2018. doi:10.1007/s41468-018-0012-6.
- 33 Vanessa Robins. Towards computing homology from finite approximations. In *Topology proceedings*, volume 24, pages 503–532, 1999.
- 34 Yuli B. Rudyak. On analytical applications of stable homotopy (the Arnold conjecture, critical points). *Mathematische Zeitschrift*, 230(4):659–672, 1999. doi:10.1007/PL00004708.
- 35 Yuli B Rudyak. On category weight and its applications. *Topology*, 38(1):37–55, 1999. doi:10.1016/S0040-9383(97)00101-8.
- 36 Parth Sarin. Cup length as a bound on topological complexity. *arXiv preprint*, 2017. arXiv:1710.06502.
- 37 Felix Schmedl. Computational aspects of the Gromov–Hausdorff distance and its application in non-rigid shape matching. *Discrete Comput. Geom.*, 57(4):854–880, June 2017. doi:10.1007/s00454-017-9889-4.
- 38 Steve Smale. On the topology of algorithms, I. *Journal of Complexity*, 3(2):81–89, 1987. doi:10.1016/0885-064X(87)90021-5.
- 39 Andrew Yarmola. *Persistence and computation of the cup product*. Undergraduate honors thesis, Stanford University, 2010.
- 40 Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005. doi:10.1007/s00454-004-1146-y.

Three-Chromatic Geometric Hypergraphs

Gábor Damásdi¹ ✉ 

MTA-ELTE Lendület Combinatorial Geometry Research Group, Dept. of Computer Science,
ELTE Eötvös Loránd University, Budapest, Hungary

Dömötör Pálvölgyi ✉ 

MTA-ELTE Lendület Combinatorial Geometry Research Group, Dept. of Computer Science,
ELTE Eötvös Loránd University, Budapest, Hungary

Abstract

We prove that for any planar convex body C there is a positive integer m with the property that any finite point set P in the plane can be three-colored such that there is no translate of C containing at least m points of P , all of the same color. As a part of the proof, we show a strengthening of the Erdős-Sands-Sauer-Woodrow conjecture. Surprisingly, the proof also relies on the two dimensional case of the Illumination conjecture.

2012 ACM Subject Classification Mathematics of computing → Hypergraphs

Keywords and phrases Discrete geometry, Geometric hypergraph coloring, Decomposition of multiple coverings

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.32

Related Version *Full Version:* <https://arxiv.org/abs/2112.01820>

Funding *Gábor Damásdi:* Supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund, and both researchers supported by the Lendület program of the Hungarian Academy of Sciences (MTA), under grant number LP2017-19/2017.

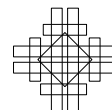
1 Introduction

Our main result is the following.

► **Theorem 1.** *For any planar convex body C there is a positive integer $m = m(C)$ such that any finite point set P in the plane can be three-colored in a way that there is no translate of C containing at least m points of P , all of the same color.*

This result closes a long line of research about coloring points with respect to planar range spaces that consist of translates of a fixed set, a problem that was initiated by Pach over forty years ago [21]. In general, a pair (P, \mathcal{S}) , where P is a set of points in the plane and \mathcal{S} is a family of subsets of the plane, called the *range space*, defines a *primal* hypergraph $\mathcal{H}(P, \mathcal{S})$ whose vertex set is P , and for each $S \in \mathcal{S}$ we add the edge $S \cap P$ to the hypergraph. Given any hypergraph \mathcal{H} , a planar realization of \mathcal{H} is defined as a pair (P, \mathcal{S}) for which $\mathcal{H}(P, \mathcal{S})$ is isomorphic to \mathcal{H} . If \mathcal{H} can be realized with some pair (P, \mathcal{S}) where \mathcal{S} is from some family \mathcal{F} , then we say that \mathcal{H} is realizable with \mathcal{F} . The dual of the hypergraph $\mathcal{H}(P, \mathcal{S})$, where the elements of the range space \mathcal{S} are the vertices and the points P define the edges such that $\{S \in \mathcal{S} \mid p \in S\}$ is an edge for every $p \in P$, is known as the *dual* hypergraph and is denoted by $\mathcal{H}(\mathcal{S}, P)$. If $\mathcal{H} = \mathcal{H}(\mathcal{S}, P)$ where \mathcal{S} is from some family \mathcal{F} , then we say that \mathcal{H}

¹ Corresponding author



has a dual realization with \mathcal{F} . Pach observed [21, 24] that if \mathcal{F} is the family of translates of some set, then \mathcal{H} has a dual realization with \mathcal{F} if and only if \mathcal{H} has a (primal) realization with \mathcal{F} .

Pach proposed to study the chromatic number of hypergraphs realizable with different geometric families \mathcal{F} . It is important to distinguish between two types of hypergraph colorings that we will use, the *proper* coloring and the *polychromatic* coloring.

► **Definition 2.** *A hypergraph is properly k -colorable if its vertices can be colored with k colors so that each edge contains points from at least two color classes. Such a coloring is called a proper k -coloring. If a hypergraph has a proper k -coloring but not a proper $(k - 1)$ -coloring, then it is called k -chromatic.*

A hypergraph is polychromatic k -colorable if its vertices can be colored with k colors so that each edge contains points from each color class. Such a coloring is called a polychromatic k -coloring.

Note that for a polychromatic k -coloring to exist, it is necessary that each edge of the underlying hypergraph has at least k vertices. More generally, we say that a hypergraph is *m -heavy* if each of its edges has at least m vertices.

The main question that Pach raised can be rephrased as follows.

► **Question 3.** *For which planar families \mathcal{F} is there an $m_k = m(\mathcal{F}, k)$ such that any m_k -heavy hypergraph realizable with \mathcal{F} has a proper/polychromatic k -coloring?*

Initially, this question has been mainly studied for polychromatic k -colorings (known in case of a dual range space as *cover-decomposition* problem), and it was shown that such an m_k exists if \mathcal{F} is the family of translates of some convex polygon [22, 33, 28], or the family of all halfplanes [14, 32], or the homothetic² copies of a triangle [15] or of a square [2], while it was also shown that even m_2 does not exist if \mathcal{F} is the family of translates of some appropriate concave polygon [26, 27] or any body³ with a smooth boundary [23]. It was also shown that there is no m_k for proper k -colorings if \mathcal{F} is the family of all lines [26] or all axis-parallel rectangles [10]; for these families, the same holds in case of dual realizations [26, 25]. For homothets of convex polygons other than triangles, it is known that there is no m_2 for dual realizations [19], unlike for primal realizations. Higher dimensional variants [15, 8] and improved bounds for m_k have been also studied [3, 13, 7, 16, 4, 9]. For other results, see also the decade old survey [24], or the up-to-date website <https://coge.elte.hu/cogezoo.html>.

If \mathcal{F} is the translates or homothets of some planar convex body, it is an easy consequence of the properties of generalized Delaunay-triangulations and the Four Color Theorem that any hypergraph realizable with \mathcal{F} is proper 4-colorable if every edge contains at least two vertices. We have recently shown that this cannot be improved for homothets.

► **Theorem 4** (Damásdi, Pálvölgyi [12]). *Let C be any convex body in the plane that has two parallel supporting lines such that C is strictly convex in some neighborhood of the two points of tangencies. For any positive integer m , there exists a 4-chromatic m -uniform hypergraph that is realizable with homothets of C .*

² A *homothetic copy*, or *homothet*, is a scaled and translated (but non-rotated) copy of a set. We always require the scaling factor to be positive. Note that this is sometimes called a positive homothet.

³ By *body*, we always mean a compact subset of the plane with a non-empty interior, though our results (and most of the results mentioned) also hold for sets that are unbounded, or that contain an arbitrary part of their boundary, and are thus neither open, nor closed. This is because a realization of a hypergraph can be perturbed slightly to move the points off from the boundaries of the sets realizing the respective edges of the hypergraph.

For translates, we recall the following result.

► **Theorem 5** (Pach, Pálvölgyi [23]). *Let C be any convex body in the plane that has two parallel supporting lines such that C is strictly convex in some neighborhood of the two points of tangencies. For any positive integer m , there exists a 3-chromatic m -uniform hypergraph that is realizable with translates of C .*

This left only the following question open: Is it true for any planar convex body C that there is a positive integer m such that no 4-chromatic m -uniform hypergraph is realizable with translates of C ? Our Theorem 1 answers this question affirmatively for all C by showing that all realizable m -heavy hypergraphs are three-colorable for some m . This has been hitherto known to hold only when C is a polygon (in which case 2 colors suffice [28], and 3 colors are known to be enough even for homothets [18]) and pseudodisk families that intersect in a common point [1] (which generalizes the case when C is unbounded, in which case 2 colors suffice [23]).

The proof of Theorem 1 relies on a surprising connection with two other famous results, the solution of the two dimensional case of the Illumination conjecture [20], and a recent solution of the Erdős-Sands-Sauer-Woodrow conjecture by Bousquet, Lochet and Thomassé [6]. In fact, we need a generalization of the latter result, which we prove with the addition of one more trick to their method; this can be of independent interest.

Note that the extended abstract of our first proof attempt appeared recently in the proceedings of EuroComb 2021 [11]. That proof did not use the above two results, however, it only worked when C was a disk, and while the generalization to other convex bodies with a smooth boundary seemed feasible, we saw no way to extend it to arbitrary convex bodies.

The rest of the paper is organized as follows.

In Section 2 we present the three main ingredients of our proof:

- the Union Lemma (Section 2.1),
- the Erdős-Sands-Sauer-Woodrow conjecture (Section 2.2) – the proof of our generalization of the Bousquet-Lochet-Thomassé theorem can be found in the full version of the paper,
- the Illumination conjecture (Section 2.3), which is a theorem of Levi in the plane.

In Section 3 we give the detailed proof of Theorem 1.

In Section 4 we give a general overview of the steps of the algorithm requiring computation to show that we can find a three-coloring in randomized polynomial time.

Finally, in Section 5, we pose some problems left open.

2 Tools

2.1 Union Lemma

Polychromatic colorability is a much stronger property than proper colorability. Any polychromatic k -colorable hypergraph is proper 2-colorable. We generalize this trivial observation to the following statement about unions of polychromatic k -colorable hypergraphs.

► **Lemma 6** (Union Lemma). *Let $\mathcal{H}_1 = (V, E_1), \dots, \mathcal{H}_{k-1} = (V, E_{k-1})$ be hypergraphs on a common vertex set V . If $\mathcal{H}_1, \dots, \mathcal{H}_{k-1}$ are polychromatic k -colorable, then the hypergraph*

$\bigcup_{i=1}^{k-1} \mathcal{H}_i = (V, \bigcup_{i=1}^{k-1} E_i)$ *is proper k -colorable.*

Proof. Choose $c(v) \in \{1, \dots, k\}$ such that it differs from each $c_i(v)$. We claim that c is a proper k -coloring of $\bigcup_{i=1}^{k-1} \mathcal{H}_i$. To prove this, it is enough to show that for every edge $H \in \mathcal{H}_i$ and for every color $j \in \{1, \dots, k-1\}$, there is a $v \in H$ such that $c(v) \neq j$. We can pick $v \in H$ for which $c_i(v) = j$. This finishes the proof. ◀

Lemma 6 is sharp in the sense that for every k there are $k-1$ hypergraphs such that each is polychromatic k -colorable but their union is not properly $(k-1)$ -colorable.

We will apply the Union Lemma combined with the theorem below. A *pseudoline arrangement* is a collection of simple curves, each of which splits \mathbb{R}^2 into two unbounded parts, such that any two curves intersect at most once. A *pseudohalfplane* is the region on one side of a pseudoline in such an arrangement. For hypergraphs realizable by pseudohalfplanes the following was proved, generalizing a result of Smorodinsky and Yuditsky [32] about halfplanes.

► **Theorem 7** (Keszegh-Pálvölgyi [17]). *Any $(2k-1)$ -heavy hypergraph realizable by pseudohalfplanes is polychromatic k -colorable, i.e., given a finite set of points and a pseudohalfplane arrangement in the plane, the points can be k -colored such that every pseudohalfplane that contains at least $2k-1$ points contains all k colors.*

Combining Theorem 7 with Lemma 6 for $k=3$, we obtain the following.

► **Corollary 8.** *Any 5-heavy hypergraph realizable by two pseudohalfplane families is proper 3-colorable, i.e., given a finite set of points and two different pseudohalfplane arrangements in the plane, the points can be 3-colored such that every pseudohalfplane that contains at least 5 points contains two differently colored points.*

2.2 Erdős-Sands-Sauer-Woodrow conjecture

Given a quasi-order⁴ \prec on a set V , we interpret it as a digraph $D = (V, A)$, where the vertex set is V and a pair (x, y) defines an arc in A if $x \prec y$. The *closed in-neighborhood* of a vertex $x \in V$ is $N^-(x) = \{x\} \cup \{y \mid (y, x) \in A\}$. Similarly the *closed out-neighborhood* of a vertex x is $N^+(x) = \{x\} \cup \{y \mid (x, y) \in A\}$. We extend this to subsets $S \subset V$ as $N^-(S) = \bigcup_{x \in S} N^-(x)$ and $N^+(S) = \bigcup_{x \in S} N^+(x)$. A set of vertices S such that $N^+(S) = V$ is said to be *dominating*.

For $A, B \subset V$ we will also say that A *dominates* B if $B \subset N^+(A)$.

A *complete multidigraph* is a digraph where parallel edges are allowed and in which there is at least one arc between each pair of distinct vertices. Let D be a complete multidigraph whose arcs are the disjoint union of k quasi-orders \prec_1, \dots, \prec_k (parallel arcs are allowed). Define $N_i^-(x)$ (resp. $N_i^+(x)$) as the closed in-neighborhood (resp. out-neighborhood) of the digraph induced by \prec_i .

Proving the conjecture of Erdős, and of Sands, Sauer and Woodrow [31], Bousquet, Lochet and Thomassé recently showed the following.

► **Theorem 9** (Bousquet, Lochet, Thomassé [6]). *For every k , there exists an integer $f(k)$ such that if D is a complete multidigraph whose arcs are the union of k quasi-orders, then D has a dominating set of size at most $f(k)$.*

⁴ A quasi-order \prec is a reflexive and transitive relation, but it is not required to be antisymmetric, so $p \prec q \prec p$ is allowed, unlike for partial orders.

We show the following generalization of Theorem 9.

► **Theorem 10.** *For every pair of positive integers k and l , there exist an integer $f(k, l)$ such that if $D = (V, A)$ is a complete multidigraph whose arcs are the union of k quasi-orders \prec_1, \dots, \prec_k , then V contains a family of pairwise disjoint subsets S_i^j for $i \in [k]$, $j \in [l]$ with the following properties:*

- $|\bigcup_{i,j} S_i^j| \leq f(k, l)$
- For each vertex $v \in V \setminus \bigcup_{i,j} S_i^j$ there is an $i \in [k]$ such that for each $j \in [l]$ there is an edge of \prec_i from a vertex of S_i^j to v .

Note that disjointness is the real difficulty here, without it the theorem would trivially hold from repeated applications of Theorem 9. We saw no way to derive Theorem 10 from Theorem 9, but with an extra modification the proof goes through. The proof of Theorem 10 can be found in the full version of the paper.

2.3 Hadwiger's Illumination conjecture and pseudolines

Hadwiger's Illumination conjecture has a number of equivalent formulations and names⁵. For a recent survey, see [5]. We will use the following version of the conjecture.

Let \mathbb{S}^{d-1} denote the unit sphere in \mathbb{R}^d . For a convex body C , let ∂C denote the boundary of C and let $\text{int}(C)$ denote its interior. A direction (light) $u \in \mathbb{S}^{d-1}$ illuminates $b \in \partial C$ if $\{b + \lambda u : \lambda > 0\} \cap \text{int}(C) \neq \emptyset$.

► **Conjecture 11.** *The boundary of any convex body in \mathbb{R}^d can be illuminated by 2^d or fewer directions. Furthermore, the 2^d lights are necessary if and only if the body is a parallelepiped.*

The conjecture is open in general. The $d = 2$ case was settled in affirmative by Levi [20] in 1955. For $d = 3$ the best result is due to Prymak [30], who showed that 16 lights are enough, improving the earlier method of Papadoperakis [29] with the help of a computer program.

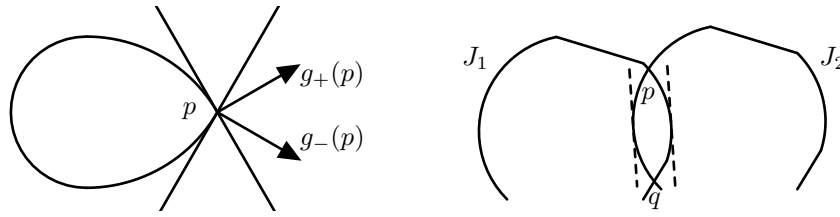
In the following part we make an interesting connection between the Illumination conjecture for $d = 2$ and pseudolines. Roughly speaking, we show that the Illumination conjecture implies that for any convex body in the plane the boundary can be broken into three parts such that the translates of each part behave similarly to pseudolines, i.e., we get three pseudoline arrangements from the translates of the three parts.

To put this into precise terms, we need some technical definitions and statements. Fix a body C and an injective parametrization of ∂C , $\gamma : [0, 1] \rightarrow \partial C$, that follows ∂C counterclockwise. For each point p of ∂C there is a set of possible tangents touching at p . Let $g(p) \subset \mathbb{S}^1$ denote the Gauss image of p , i.e., $g(p)$ is the set of unit outernormals of the tangent lines touching at p . Note that $g(p)$ is an arc of \mathbb{S}^1 and $g(p)$ is a proper subset of \mathbb{S}^1 .

Let $g_+ : \partial C \rightarrow \mathbb{S}^1$ be the function that assigns to p the counterclockwise last element of $g(p)$. (See Figure 1 left.) Similarly let g_- be the function that assigns to p the clockwise last element of $g(p)$. Thus, $g(p)$ is the arc of \mathbb{S}^1 from $g_-(p)$ to $g_+(p)$. Let $|g(p)|$ denote the length of $g(p)$.

► **Observation 12.** $g_+ \circ \gamma$ is continuous from the right and $g_- \circ \gamma$ is continuous from the left.

⁵ These include names such as Levi–Hadwiger Conjecture, Gohberg–Markus Covering Conjecture, Hadwiger Covering Conjecture, Boltyanski–Hadwiger Illumination Conjecture.



■ **Figure 1** Extremal tangents at a boundary point (on the left) and parallel tangents on two intersecting translates (on the right).

For $t_1 < t_2$ let $\gamma_{[t_1, t_2]}$ denote the restriction of γ to the interval $[t_1, t_2]$. For $t_1 > t_2$ let $\gamma_{[t_1, t_2]}$ denote the concatenation of $\gamma_{[t_1, 1]}$ and $\gamma_{[0, t_2]}$. When it leads to no confusion, we identify $\gamma_{[t_1, t_2]}$ with its image, which is a closed connected part of the boundary ∂C . For such a $J = \gamma_{[t_1, t_2]}$, let $g(J) = \bigcup_{p \in J} g(p)$. Clearly, $g(J)$ is an arc of \mathbb{S}^1 from $g_-(t_1)$ to $g_+(t_2)$; let $|g(J)|$ denote the length of this arc.

► **Lemma 13.** *Let C be a convex body and assume that J is a closed connected part of ∂C such that $|g(J)| < \pi$. Then there are no two translates of J that intersect in more than one point.*

Proof. Suppose J has two translates J_1 and J_2 such that they intersect in two points, p and q . Now both J_1 and J_2 have a tangent that is parallel to the segment pq , but since they lie on different sides of the pq line, they have opposite outer normal vectors. (See Figure 1 right.) This shows that J has two different tangents parallel to pq and therefore $|g(J)| \geq \pi$. ◀

► **Lemma 14.** *For a convex body C , which is not a parallelogram, and an injective parametrization γ of ∂C , we can pick $0 \leq t_1 < t_2 < t_3 \leq 1$ such that $|g(\gamma_{[t_1, t_2]})|$, $|g(\gamma_{[t_2, t_3]})|$ and $|g(\gamma_{[t_3, t_1]})|$ are each strictly smaller than π .*

Proof. We use the 2-dimensional case of the Illumination conjecture (proved by Levi [20]). If C is not a parallelogram, we can pick three directions, u_1, u_2 and u_3 , that illuminate C . Pick t_1 such that $\gamma(t_1)$ is illuminated by both u_1 and u_2 . To see why this is possible, suppose that the parts illuminated by u_1 and u_2 are disjoint. Each light illuminates a continuous open ended part of the boundary. So in this case there are two disjoint parts of the boundary that are not illuminated. If u_3 illuminates both, then it illuminates everything that is illuminated by u_1 or everything that is illuminated by u_2 . This would mean that two lights illuminate the whole boundary but this is not possible for any convex body. Indeed, suppose that two lights u and v illuminate the whole body. Then there is a halfplane H through the origin that contains both vectors u and v . Take a translate of H that touches C . Clearly the touching point is not illuminated by either u or v , a contradiction.

Using the same argument, pick t_2 and t_3 such that $\gamma(t_2)$ is illuminated by both u_2 and u_3 and $\gamma(t_3)$ is illuminated by both u_3 and u_1 .

Note that u_1 illuminates exactly those points for which $g_+(p) < u_1 + \pi/2$ and $g_-(p) > u_1 - \pi/2$. Therefore, $|g(\gamma_{[t_1, t_3]})| < u_1 + \pi/2 - (u_1 - \pi/2) = \pi$. Similarly $|g(\gamma_{[t_1, t_2]})| < \pi$ and $|g(\gamma_{[t_2, t_3]})| < \pi$. ◀

Observation 12 and Lemma 14 immediately imply the following statement.

► **Lemma 15.** *For a convex body C , which is not a parallelogram, and an injective parametrization γ of ∂C , we can pick $0 \leq t_1 < t_2 < t_3 \leq 1$ and $\varepsilon > 0$ such that $|g(\gamma_{[t_1 - \varepsilon, t_2 + \varepsilon]})|$, $|g(\gamma_{[t_2 - \varepsilon, t_3 + \varepsilon]})|$ and $|g(\gamma_{[t_3 - \varepsilon, t_1 + \varepsilon]})|$ are each strictly smaller than π .*

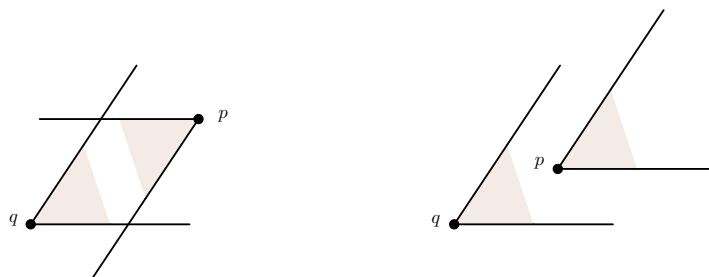
3 Proof of Theorem 1

3.1 Quasi-orders on planar point sets

Cones provide a natural way to define quasi-orders on point sets (see [33] for an example where this idea was used). A *cone* is a closed region in the plane that is bounded by two rays that emanate from the origin. For a cone K let $-K$ denote the cone that is the reflection of K across the origin and let $q + K$ denote the translate of K by the vector q .

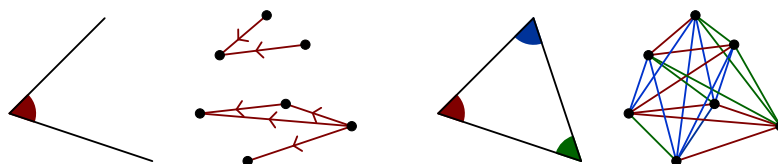
► **Observation 16.** For any $p, q \in \mathbb{R}^2$ and cone K , the following are equivalent (see Fig. 2):

- $p \in q + K$
- $q \in p + (-K)$
- $p + K \subseteq q + K$



■ **Figure 2** Basic properties of cones.

For a cone K let \prec_K denote the relation on the points of the plane where a point p is bigger than a point q if and only if $p + K$ contains q . By Observation 16, this relation is transitive so it is a quasi-order. Recall that when \prec_K is interpreted as a digraph, qp is an edge if and only if $q \prec_K p$.



■ **Figure 3** Quasi-order on a point set.

Suppose the cones K_1, K_2, K_3 are the translates of the three corners of a triangle so that all their apexes are in the origin, in other words the cones $K_1, -K_3, K_2, -K_1, K_3, -K_2$ partition the plane around the origin in this order. Then we will say that K_1, K_2, K_3 is a *set of tri-partition cones*. In this case the intersection of any translates of K_1, K_2, K_3 forms a (possibly degenerate) triangle.

► **Observation 17.** Let K_1, K_2, K_3 be a set of tri-partition cones and let P be a planar point set. Then any two distinct points of P are comparable in either \prec_{K_1}, \prec_{K_2} or \prec_{K_3} . (See Figure 3.)

In other words, when interpreted as digraphs, the union of \prec_{K_1}, \prec_{K_2} and \prec_{K_3} forms a complete multidigraph on P . As a warm up for the proof of Theorem 1, we show the following theorem.

► **Theorem 18.** *There exists a positive integer m such that for any point set P , and any set of tri-partition cones K_1, K_2, K_3 , we can three-color P such that no translate of K_1, K_2 or K_3 that contains at least m points of P is monochromatic.*

Proof. We set m to be $f(3, 2) + 13$ with the function of Theorem 10. Consider the three quasi-orders \prec_{K_1}, \prec_{K_2} or \prec_{K_3} . Their union gives a complete multidigraph on P , hence we can apply Theorem 10 with $k = 3$ and $l = 2$, resulting in subsets S_i^j for $i \in [3], j \in [2]$. Let $S = \bigcup_{i \in [3], j \in [2]} S_i^j$. For each point $p \in P \setminus S$ there is an i such that \prec_{K_i} has an edge from a vertex of $S_{i,1}$ and $S_{i,2}$ to p . Let P_1, P_2, P_3 be the partition of $P \setminus S$ according to this i value.

We start by coloring the points of S . Color the points of $S_{1,1} \cup S_{2,1} \cup S_{3,1}$ with the first color and color the points of $S_{1,2} \cup S_{2,2} \cup S_{3,2}$ with the second color.

Any translate of K_1, K_2 or K_3 that contains $f(3, 2) + 13$ points of P , must contain 5 points from either P_1, P_2 or P_3 by the pigeonhole principle. (Note that the cone might contain all points of S .) Therefore, it is enough to show that for each $i \in [3]$ the points of P_i can be three-colored such that no translate of K_1, K_2 , or K_3 that contains at least 5 points of P_i is monochromatic.

Consider P_1 ; the proof is the same for P_2 and P_3 . Take a translate of K_1 and suppose that it contains a point p of P_1 . By Theorem 10, there is an edge of \prec_{K_1} from a vertex of $S_{1,1}$ to p and another edge from a vertex of $S_{1,2}$ to p . Thus any such translate contains a point from $S_{1,1}$ and another point from $S_{1,2}$, and hence it cannot be monochromatic.

Therefore, we only have to consider the translates of K_2 and K_3 . Two translates of a cone intersect at most once on their boundary. Hence, the translates of K_2 form a pseudohalfplane arrangement, and so do the translates of K_3 . Therefore, by Corollary 8, there is a proper three-coloring for the translates of K_2 and K_3 together. ◀

► **Remark 19.** From Theorem 18, it follows using standard methods (see Section 3.2) that Theorem 1 holds for triangles. This was of course known before, even for two-colorings of homothetic copies of triangles. Our proof cannot be modified for homothets, but a two-coloring would follow if instead of Corollary 8 we applied a more careful analysis for the two cones.

3.2 Proof of Theorem 1

If C is a parallelogram, then our proof method fails. Luckily, translates of parallelograms (and other symmetric polygons) were the first for which it was shown that even two colors are enough [22]; in fact, by now we know that two colors are enough even for homothets of parallelograms [2]. So from now on we assume that C is not a parallelogram.

The proof of Theorem 1 relies on the same ideas as we used for Theorem 18. We partition P into several parts, and for each part P_i , we divide the translates of C into three families such that two of the families each form a pseudohalfplane arrangement over P_i , while the third family will only contain translates that are automatically non-monochromatic. Then Corollary 8 gives us a proper three-coloring. As in the proof of Theorem 18, this is not done directly. First, we divide the plane using a grid, and then in each small square we will use Theorem 10 to discard some of the translates of C at the cost of a bounded number of points.

Now we start the proof of Theorem 1. The first step is a classic divide and conquer idea [22]. We chose a constant $r = r(C)$ depending only on C and divide the plane into a grid of squares of side length r . Since each translate of C intersects some bounded number of squares, by the pigeonhole principle we can find for any positive integer m another integer m' such that the following holds: each translate \hat{C} of C that contains at least m' points

intersects a square Q such that $\hat{C} \cap Q$ contains at least m points. For example, we can choose $m' = m(\text{diam}(C)/r + 2)^2$, where $\text{diam}(C)$ denotes the diameter of C . Therefore, it is enough to show the following localized version of Theorem 1, since applying it separately for the points in each square of the grid provides a proper three-coloring of the whole point set.

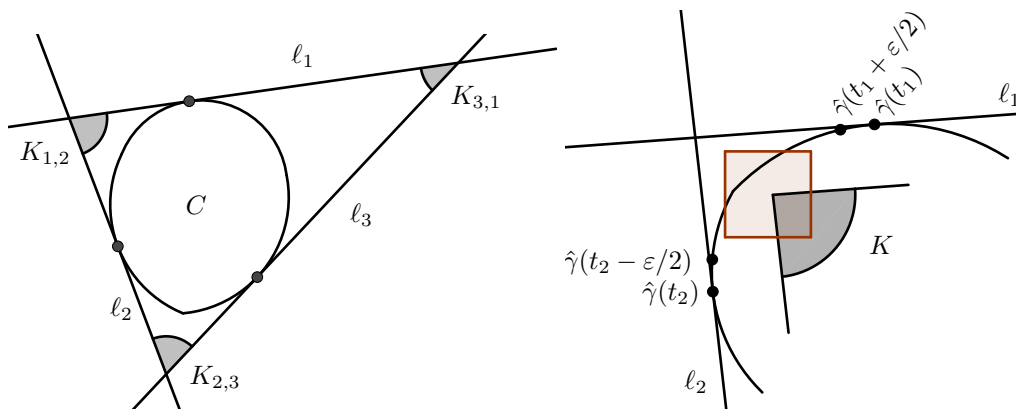
► **Theorem 20.** *There is a positive integer m such that for any convex body C there is a positive real r such that any finite point set P in the plane that lies in a square of side length r can be three-colored in a way that there is no translate of C containing at least m points of P , all of the same color.*

We will show that m can be chosen to be $f(3, 2) + 13$ with the function of Theorem 10, independently of C .

Proof. We pick r the following way. First we fix an injective parametrization γ of ∂C and then fix t_1, t_2, t_3 and ε according to Lemma 15. Let ℓ_1, ℓ_2, ℓ_3 be the tangents of C touching at $\gamma(t_1), \gamma(t_2)$ and $\gamma(t_3)$. Let $K_{1,2}, K_{2,3}, K_{3,1}$ be the set of tri-partition cones bordered by ℓ_1, ℓ_2, ℓ_3 , such that $K_{i,i+1}$ is bordered by ℓ_i on its counterclockwise side, and by ℓ_{i+1} on its clockwise side (see Figure 4 left, and note that we always treat $3 + 1$ as 1 in the subscript).

For a translate \hat{C} of C we will denote by $\hat{\gamma}$ the translated parametrization of $\partial \hat{C}$, i.e., $\hat{\gamma}(t) = \gamma(t) + v$ if \hat{C} was translated by vector v . Our aim is to choose r small enough to satisfy the following two properties for each $i \in [3]$.

- (A) Let \hat{C} be a translate of C , and Q be a square of side length r such that $\partial \hat{C} \cap Q \subset \hat{\gamma}_{[t_i + \varepsilon/2, t_{i+1} - \varepsilon/2]}$ (see Figure 4 right). Then for any translate K of $K_{i,i+1}$ whose apex is in $Q \cap \hat{C}$, we have $K \cap Q \subset \hat{C}$. (I.e., r is small with respect to C .)
- (B) Let \hat{C} be a translate of C , and Q be a square of side length r such that $\hat{\gamma}_{[t_i - \varepsilon/2, t_{i+1} + \varepsilon/2]}$ intersects Q . Then $\partial \hat{C} \cap Q \subset \hat{\gamma}_{[t_i - \varepsilon, t_{i+1} + \varepsilon]}$. (I.e., r is small compared to ε .)



■ **Figure 4** Selecting the cones (on the left) and Property (A) (on the right).

We show that an r satisfying properties (A) and (B) can be found for $i = 1$. The argument is the same for $i = 2$ and $i = 3$, and we can take the smallest among the three resulting values of r .

First, consider property (A). Since the sides of K are parallel to ℓ_1 and ℓ_2 , the portion of K that lies “above” the segment $\overline{\hat{\gamma}(t_1)\hat{\gamma}(t_2)}$ is in \hat{C} . Hence, if we choose r small enough so that Q cannot intersect $\overline{\hat{\gamma}(t_1)\hat{\gamma}(t_2)}$, then property (A) is satisfied. We can choose r to be smaller than $\frac{1}{\sqrt{2}}$ times the distance of the segments $\overline{\hat{\gamma}(t_1)\hat{\gamma}(t_2)}$ and $\overline{\hat{\gamma}(t_1 + \varepsilon/2)\hat{\gamma}(t_2 - \varepsilon/2)}$.

Using that γ is a continuous function on a compact set, we can pick r such that property (B) is satisfied. Therefore, there is an r satisfying properties (A) and (B).

The next step is a subdivision of the point set P using Theorem 10, like we did in the proof of Theorem 18. The beginning of our argument is exactly the same.

Apply Theorem 10 for the graph given by the union of $\prec_{K_{1,2}}$, $\prec_{K_{2,3}}$ and $\prec_{K_{3,1}}$. By Observation 16, this is indeed a complete multidigraph on P .

We apply Theorem 10 with $k = 3$ and $l = 2$, resulting in subsets S_i^j for $i \in [3], j \in [2]$. Let $S = \bigcup_{i \in [3], j \in [2]} S_i^j$. For each point $p \in P \setminus S$ there is an i such that $\prec_{K_{i,i+1}}$ has an edge from a vertex of $S_{i,1}$ and $S_{i,2}$ to p . Let P_1, P_2, P_3 be the partition of $P \setminus S$ according to this i value.

We start by coloring the points of S . Color the points of $S_{1,1} \cup S_{2,1} \cup S_{3,1}$ with the first color and color the points of $S_{1,2} \cup S_{2,2} \cup S_{3,2}$ with the second color.

Note that m is at least $f(3, 2) + 13$. Any translate of C that contains $f(3, 2) + 13$ points of P must contain 5 points from either P_1, P_2 or P_3 . (Note that the cone might contain all points of S). Thus, it is enough to show that for each $i \in [3]$ the points of P_i can be 3-colored so that no translate of C that contains at least 5 points of P_i is monochromatic.

Consider P_1 , the proof is the same for P_2 and P_3 . We divide the translates of C that intersect Q into four (not necessarily disjoint) groups. Let \mathcal{C}_0 denote the translates where $\hat{C} \cap Q = Q$. Let \mathcal{C}_1 denote the translates for which $\partial\hat{C} \cap Q \subset \hat{\gamma}_{[t_1+\varepsilon/2, t_2-\varepsilon/2]}$. Let \mathcal{C}_2 denote the translates for which $\partial\hat{C} \cap Q \cap \hat{\gamma}_{[t_2-\varepsilon/2, t_3]}$ $\neq \emptyset$. Let \mathcal{C}_3 denote the remaining translates for which $\partial\hat{C} \cap Q \cap \hat{\gamma}_{[t_3, t_1+\varepsilon/2]}$ $\neq \emptyset$.

We do not need to worry about the translates in \mathcal{C}_0 , as Q itself will not be monochromatic.

Take a translate \hat{C} from \mathcal{C}_1 and suppose that it contains a point $p \in P_1$. By Theorem 10, there is an edge of $\prec_{K_{1,2}}$ from a vertex of $S_{1,1}$ to p and another edge from a vertex of $S_{1,2}$ to p . I.e., the cone $p + K_{1,2}$ contains a point from $S_{1,1}$ and another point from $S_{1,2}$, and hence it is not monochromatic. From property (A) we know that every point in $(p + K_{1,2}) \cap P$ is also in \hat{C} . Therefore, \hat{C} is not monochromatic.

Now consider the translates in \mathcal{C}_2 . From property (B) we know that for these translates we have $\partial\hat{C} \cap Q \subset \hat{\gamma}_{[t_2-\varepsilon, t_3+\varepsilon]}$. By the definition of t_1, t_2 and t_3 , we know that this implies that any two translates from \mathcal{C}_2 intersect at most once on their boundary within Q , i.e., they behave as pseudohalfplanes. To turn the translates in \mathcal{C}_2 into a pseudohalfplane arrangement as defined earlier, we can do as follows. For a translate \hat{C} , replace it with the convex set whose boundary is $\hat{\gamma}_{[t_2-\varepsilon, t_3+\varepsilon]}$ extended from its endpoints with two rays orthogonal to the segment $\hat{\gamma}(t_2 - \varepsilon)\hat{\gamma}(t_3 + \varepsilon)$. This new family provides the same intersection pattern in Q and forms a pseudohalfplane arrangement. We can do the same with the translates in \mathcal{C}_3 . Therefore, by Corollary 8 there is a proper three-coloring for the translates in $\mathcal{C}_2 \cup \mathcal{C}_3$. \blacktriangleleft

4 Overview of the computational complexity of the algorithm

In this section we show that given a point set P and a convex set C , we can determine some $m = m(C)$ and calculate a three-coloring of P efficiently if C is given in a natural way, for example, if C is a disk. Our algorithm is randomized and its expected running time is a polynomial of the number of points, $n = |P|$.

- First, we need to fix three points on the boundary, $\tau_1, \tau_2, \tau_3 \subset \partial C$ such that Lemma 15 is satisfied with $\tau_i = \gamma(t_i)$ for some t_i and $\varepsilon > 0$ for each i . Note that we do not need to fix a complete parametrization γ of ∂C or $\varepsilon > 0$; instead, it is enough to choose some points τ_i^{--} and τ_i^{++} that satisfy the conclusion of Lemma 15 if we assume $\tau_i^{--} = \gamma(t_i - \varepsilon)$ and $\tau_i^{++} = \gamma(t_i + \varepsilon)$ for each i . If C has a smooth boundary, like a disk, we can pick τ_1, τ_2, τ_3 to be the touching points of an equilateral triangle with C inscribed in it. If the boundary

of C contains vertex-type sharp turns, the complexity of finding these turns depends on how C is given, but for any reasonable input method, this should be straight-forward. After that, one can follow closely the steps of the proof of the Illumination conjecture in the plane to get an algorithm, but apparently, this has not yet been studied in detail.

- To pick r , the side length of the squares of the grid, we can fix some arbitrary points τ_i^- between τ_i^{--} and τ_i , and points τ_i^+ between τ_i and τ_i^{++} , to play the roles of $\gamma(t_i - \varepsilon/2)$ and $\gamma(t_i + \varepsilon/2)$, respectively, for each i . It is sufficient to pick r so that $r\sqrt{2}$, the diameter of the square of side length r , is less than
 - the distance of τ_i^+ and τ_{i+1}^- from the segment $\overline{\tau_i\tau_{i+1}}$,
 - the distance of τ_i^- from τ_i^{--} , and
 - the distance of τ_i^+ from τ_i^{++} ,
 for each i , to guarantee that properties (A) and (B) are satisfied.
- Set $m = f(3, 2) + 13$, which is an absolute constant given by Theorem 10. We need to construct the complete multidigraph given by the tri-partition cones determined by τ_1, τ_2, τ_3 , which needs a comparison for each pair of points. To obtain the subsets $S_i^j \subset P$ for $i \in [3], j \in [2]$, where P is the set of points that are contained in a square of side length r , we randomly sample the required number of points from each of the constantly many T_{j_1, \dots, j_i} according to the probability distributions w_{j_1, \dots, j_i} given in the proof. These probability distributions can be computed by LP. With high probability, all the S_i^j -s will be disjoint – otherwise, we can resample until we obtain disjoint sets.
- To find the three-coloring for the two pseudohalfplane arrangements, given by Corollary 8, it is enough to determine the two-coloring given by Theorem 7 for one pseudohalfplane arrangement. While not mentioned explicitly in [17], the polychromatic k -coloring can be found in polynomial time if we know the hypergraph determined by the range space, as this hypergraph can only have a polynomial number of edges, and the coloring algorithm only needs to check some simple relations among a constant number of vertices and edges.
- Finally, to compute a suitable m' for Theorem 1 from the m of Theorem 20, it is enough to know any upper bound B for the diameter of C , and let $m' = m(B/r + 2)^2$.

5 Open questions

It is a natural question whether there is a universal m that works for all convex bodies in Theorem 1, like in Theorem 20. This would follow if we could choose r to be a universal constant. While the r given by our algorithm can depend on C , we can apply an appropriate affine transformation to C before choosing r ; this does not change the hypergraphs that can be realized with the range space determined by the translates of C . To ensure that properties (A) and (B) are satisfied would require further study of the Illumination conjecture.

Our bound for m is quite large, even for the unit disk, both in Theorems 1 and 20, which is mainly due to the fact that $f(3, 2)$ given by Theorem 10 is huge. It has been conjectured that in Theorem 9 the optimal value is $f(3) = 3$, and a similarly small number seems realistic for $f(3, 2)$ as well.

While Theorem 1 closed the last question left open for primal hypergraphs realizable by translates of planar bodies, the respective problem is still open in higher dimensions. While it is not hard to show that some hypergraphs with high chromatic number often used in constructions can be easily realized by unit balls in \mathbb{R}^5 , we do not know whether the chromatic number is bounded or not in \mathbb{R}^3 . From our Union Lemma (Lemma 6) it follows that to establish boundedness, it would be enough to find a polychromatic k -coloring for pseudohalfspaces, whatever this word means.

References

- 1 Eyal Ackerman, Balázs Keszegh, and Dömötör Pálvölgyi. Coloring hypergraphs defined by stabbed pseudo-disks and ABAB-free hypergraphs. *SIAM J. Discrete Math.*, 34(4):2250–2269, 2020. doi:10.1137/19M1290231.
- 2 Eyal Ackerman, Balázs Keszegh, and Máté Vizer. Coloring points with respect to squares. *Discrete Comput. Geom.*, 58(4):757–784, 2017. doi:10.1007/s00454-017-9902-y.
- 3 Greg Aloupis, Jean Cardinal, Sébastien Collette, Stefan Langerman, David Orden, and Pedro Ramos. Decomposition of multiple coverings into more parts. *Discrete Comput. Geom.*, 44(3):706–723, 2010. doi:10.1007/s00454-009-9238-3.
- 4 Andrei Asinowski, Jean Cardinal, Nathann Cohen, Sébastien Collette, Thomas Hackl, Michael Hoffmann, Kolja Knauer, Stefan Langerman, Michał Lasoń, Piotr Micek, Günter Rote, and Torsten Ueckerdt. Coloring hypergraphs induced by dynamic point sets and bottomless rectangles. In *Algorithms and data structures*, volume 8037 of *Lecture Notes in Comput. Sci.*, pages 73–84. Springer, Heidelberg, 2013. doi:10.1007/978-3-642-40104-6_7.
- 5 Károly Bezdek and Muhammad A. Khan. The geometry of homothetic covering and illumination. In *Discrete geometry and symmetry*, volume 234 of *Springer Proc. Math. Stat.*, pages 1–30. Springer, Cham, 2018. doi:10.1007/978-3-319-78434-2_1.
- 6 Nicolas Bousquet, William Lochet, and Stéphan Thomassé. A proof of the Erdős-Sands-Sauer-Woodrow conjecture. *J. Combin. Theory Ser. B*, 137:316–319, 2019. doi:10.1016/j.jctb.2018.11.005.
- 7 Jean Cardinal, Kolja Knauer, Piotr Micek, and Torsten Ueckerdt. Making triangles colorful. *J. Comput. Geom.*, 4(1):240–246, 2013. doi:10.20382/jocg.v4i1a10.
- 8 Jean Cardinal, Kolja Knauer, Piotr Micek, and Torsten Ueckerdt. Making octants colorful and related covering decomposition problems. *SIAM J. Discrete Math.*, 28(4):1948–1959, 2014. doi:10.1137/140955975.
- 9 Jean Cardinal, Piotr Micek, Kolja Knauer, Dömötör Pálvölgyi, Torsten Ueckerdt, and Narmada Varadarajan. Colouring bottomless rectangles and arborescences. *To appear*, 2020.
- 10 Xiaomin Chen, János Pach, Mario Szegedy, and Gábor Tardos. Delaunay graphs of point sets in the plane with respect to axis-parallel rectangles. *Random Structures Algorithms*, 34(1):11–23, 2009. doi:10.1002/rsa.20246.
- 11 Gábor Damásdi and Dömötör Pálvölgyi. Unit disks hypergraphs are three-colorable. *Extended Abstracts EuroComb 2021, Trends in Mathematics*, 14:483–489, 2021.
- 12 Gábor Damásdi and Dömötör Pálvölgyi. Realizing an m -uniform four-chromatic hypergraph with disks. *Combinatorica*, to appear, 2022.
- 13 Matt Gibson and Kasturi Varadarajan. Optimally decomposing coverings with translates of a convex polygon. *Discrete Comput. Geom.*, 46(2):313–333, 2011. doi:10.1007/s00454-011-9353-9.
- 14 Balázs Keszegh. Coloring half-planes and bottomless rectangles. *Comput. Geom.*, 45(9):495–507, 2012. doi:10.1016/j.comgeo.2011.09.004.
- 15 Balázs Keszegh and Dömötör Pálvölgyi. Octants are cover-decomposable. *Discrete Comput. Geom.*, 47(3):598–609, 2012. doi:10.1007/s00454-011-9377-1.
- 16 Balázs Keszegh and Dömötör Pálvölgyi. Convex polygons are self-coverable. *Discrete Comput. Geom.*, 51(4):885–895, 2014. doi:10.1007/s00454-014-9582-9.
- 17 Balázs Keszegh and Dömötör Pálvölgyi. An abstract approach to polychromatic coloring: shallow hitting sets in ABA-free hypergraphs and pseudohalfplanes. *J. Comput. Geom.*, 10(1):1–26, 2019. doi:10.20382/jocg.v10i1a1.
- 18 Balázs Keszegh and Dömötör Pálvölgyi. Proper coloring of geometric hypergraphs. *Discrete Comput. Geom.*, 62(3):674–689, 2019. doi:10.1007/s00454-019-00096-9.
- 19 István Kovács. Indecomposable coverings with homothetic polygons. *Discrete Comput. Geom.*, 53(4):817–824, 2015. doi:10.1007/s00454-015-9687-9.
- 20 F. W. Levi. Überdeckung eines Eibereiches durch Parallelverschiebung seines offenen Kerns. *Arch. Math. (Basel)*, 6:369–370, 1955. doi:10.1007/BF01900507.

- 21 János Pach. Decomposition of multiple packing and covering. *Diskrete Geometrie, 2. Kolloq. Math. Inst. Univ. Salzburg*, pages 169–178, 1980.
- 22 János Pach. Covering the plane with convex polygons. *Discrete Comput. Geom.*, 1(1):73–81, 1986. doi:10.1007/BF02187684.
- 23 János Pach and Dömötör Pálvölgyi. Unsplittable coverings in the plane. *Adv. Math.*, 302:433–457, 2016. doi:10.1016/j.aim.2016.07.011.
- 24 János Pach, Dömötör Pálvölgyi, and Géza Tóth. Survey on decomposition of multiple coverings. In *Geometry—intuitive, discrete, and convex*, volume 24 of *Bolyai Soc. Math. Stud.*, pages 219–257. János Bolyai Math. Soc., Budapest, 2013. doi:10.1007/978-3-642-41498-5_9.
- 25 János Pach and Gábor Tardos. Coloring axis-parallel rectangles. *J. Combin. Theory Ser. A*, 117(6):776–782, 2010. doi:10.1016/j.jcta.2009.04.007.
- 26 János Pach, Gábor Tardos, and Géza Tóth. Indecomposable coverings. In *Discrete geometry, combinatorics and graph theory*, volume 4381 of *Lecture Notes in Comput. Sci.*, pages 135–148. Springer, Berlin, 2007. doi:10.1007/978-3-540-70666-3_15.
- 27 Dömötör Pálvölgyi. Indecomposable coverings with concave polygons. *Discrete Comput. Geom.*, 44(3):577–588, 2010. doi:10.1007/s00454-009-9194-y.
- 28 Dömötör Pálvölgyi and Géza Tóth. Convex polygons are cover-decomposable. *Discrete Comput. Geom.*, 43(3):483–496, 2010. doi:10.1007/s00454-009-9133-y.
- 29 Ioannis Papadoperakis. An estimate for the problem of illumination of the boundary of a convex body in E^3 . *Geom. Dedicata*, 75(3):275–285, 1999. doi:10.1023/A:1005056207406.
- 30 A Prymak. Every 3-dimensional convex body can be covered by 14 smaller homothetic copies. *arXiv preprint*, 2021. arXiv:2112.10698.
- 31 Bill Sands, Norbert W. Sauer, and Robert E. Woodrow. On monochromatic paths in edge-coloured digraphs. *J. Combin. Theory Ser. B*, 33(3):271–275, 1982. doi:10.1016/0095-8956(82)90047-8.
- 32 Shakhar Smorodinsky and Yelena Yuditsky. Polychromatic coloring for half-planes. *J. Combin. Theory Ser. A*, 119(1):146–154, 2012. doi:10.1016/j.jcta.2011.07.001.
- 33 Gábor Tardos and Géza Tóth. Multiple coverings of the plane with triangles. *Discrete Comput. Geom.*, 38(2):443–450, 2007. doi:10.1007/s00454-007-1345-4.

A Solution to Ringel’s Circle Problem

James Davies ✉

University of Waterloo, Canada

Chaya Keller ✉ 

Ariel University, Israel

Linda Kleist ✉ 

Technische Universität Braunschweig, Germany

Shakhar Smorodinsky ✉ 

Ben-Gurion University of the Negev, Beer-Sheva, Israel

Bartosz Walczak ✉ 

Department of Theoretical Computer Science, Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland

Abstract

We construct families of circles in the plane such that their tangency graphs have arbitrarily large girth and chromatic number. This provides a strong negative answer to Ringel’s circle problem (1959). The proof relies on a (multidimensional) version of Gallai’s theorem with polynomial constraints, which we derive from the Hales-Jewett theorem and which may be of independent interest.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases circle arrangement, chromatic number, Gallai’s theorem, polynomial method

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.33

Related Version *Full Version:* <https://arxiv.org/abs/2112.05042>

Funding *Chaya Keller:* Research partially supported by the Israel Science Foundation (grant no. 1065/20).

Shakhar Smorodinsky: Research partially supported by the Israel Science Foundation (grant no. 1065/20).

Bartosz Walczak: The author is partially supported by the National Science Center of Poland grant 2019/34/E/ST6/00443.

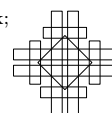
Acknowledgements This work was initiated at the online workshop “Geometric graphs and hyper-graphs”. We thank the organizers Torsten Ueckerdt and Yelena Yuditsky for a very nice workshop and all participants for fun coffee breaks and a fruitful atmosphere.

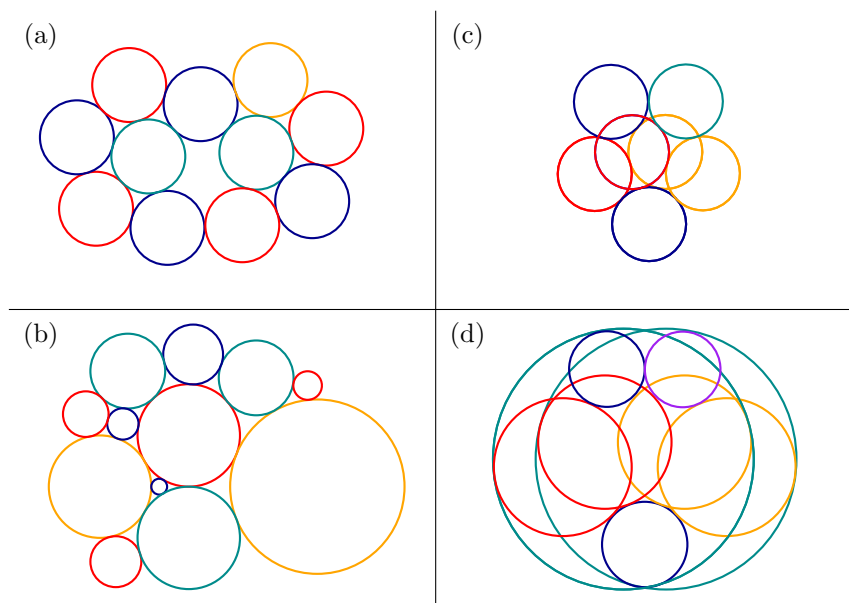
1 Introduction

A *constellation* (see [10]) is a finite collection of circles in the plane in which no three circles are tangent at the same point. The *tangency graph* $G(\mathcal{C})$ of a constellation \mathcal{C} is the graph with vertex set \mathcal{C} and edges comprising of the pairs of tangent circles in \mathcal{C} . In this paper, graph-theoretic terms such as chromatic number or girth (i.e., the minimum length of a cycle) applied to a constellation \mathcal{C} refer to the tangency graph $G(\mathcal{C})$.

Jackson and Ringel [10] discussed four problems regarding the chromatic number of constellations. The problems are illustrated in Figure 1.

- (a) *The penny problem.* What is the maximum chromatic number of a constellation of non-overlapping unit circles?
- (b) *The coin problem.* What is the maximum chromatic number of a constellation of non-overlapping circles (of arbitrary radii)?





■ **Figure 1** An illustration of the four coloring problems of tangency graphs of constellations: (a) a penny graph, (b) a coin graph, (c) an overlapping penny graph, and (d) a general constellation as in the circle problem.

- (c) *The overlapping penny problem.* What is the maximum chromatic number of a (possibly overlapping) constellation of unit circles?
- (d) *The circle problem.* What is the maximum chromatic number of a general constellation of circles?

Jackson and Ringel provided a simple proof that the answer to the *penny problem* is 4. The claim that the answer for the *coin problem* is also 4 is equivalent to the *four color theorem* [1, 2]. Indeed, on the one hand, if the circles are non-overlapping, then $G(\mathcal{C})$ is planar and thus 4-colorable by the four-color theorem. On the other hand, by the Koebe-Andreiev-Thurston *circle packing theorem* [13], every planar graph can be realized as $G(\mathcal{C})$ for some constellation \mathcal{C} of non-overlapping circles, and hence, the assertion that every such constellation \mathcal{C} is 4-colorable implies the four color theorem.

The *overlapping penny problem* is equivalent to the celebrated Hadwiger-Nelson problem, which asks what is the minimum number of colors needed for a coloring of the plane such that no two points at distance 1 get the same color. Indeed, if all circles in \mathcal{C} have a radius of $1/2$, then two circles are tangent if and only if the distance between their centers is 1. For this setting, Isbell [20] observed about 60 years ago that 7 colors suffice, and only recently de Grey [5] showed that 4 colors are not sufficient, and hence, the chromatic number of the plane lies between 5 and 7.

Unlike for the first three problems, in which a finite upper bound was known already when they were stated, for the *circle problem* no finite upper bound was known. This open problem was introduced for the first time by Ringel [19] in 1959 and appeared in several places as either a question (e.g., [10, 11, 15]) or a conjecture that there is a finite upper bound (e.g., [12]). For lower bounds, Jackson and Ringel [10] presented an example that requires 5 colors, see Figure 1(d). Another such example follows from de Grey's 5-chromatic unit distance graph. No construction requiring more than 5 colors has been known so far.

In this paper, we solve Ringel's circle problem in a strong sense by showing that the chromatic number is unbounded, even if we require high girth.

► **Theorem 1.** *There exist constellations of circles in the plane with arbitrarily large girth and chromatic number.*

The constellation condition (that no three circles are tangent at a point) is crucial for Ringel’s circle problem to be interesting – otherwise one could drive the chromatic number arbitrarily high by taking a set of circles all tangent at one point. In Theorem 1, however, the condition is redundant because it follows from the stronger condition that the girth of the tangency graph is larger than 3. Actually, we prove an even stronger statement (Theorem 9) in which we additionally forbid pairs of internally tangent circles.

The first author [4] recently proved that there are intersection graphs of axis-aligned boxes in \mathbb{R}^3 with arbitrarily large girth and chromatic number. The main tool for this result is a “sparse” version of Gallai’s theorem due to Prömel and Voigt [17] (see Theorem 3), whose applications include a modification of Tutte’s construction of triangle-free graphs with large chromatic number [6, 7].

To prove Theorem 1, we also use a “sparse” version of Gallai’s theorem. However, it is crucial in our context to guarantee that there are no “unwanted” tangencies in the resulting collection of circles. To this end, we develop a refined “sparse” version of Gallai’s theorem with additional (polynomial) constraints (Theorem 4). We believe that this version may be applicable to obtaining lower bound constructions for other geometric coloring problems, in which some specific form of algebraic independence is requested.

Tangent circles can be thought of as circles intersecting at zero angle. We extend Theorem 1 to graphs defined by pairs of circles intersecting at an arbitrary fixed angle. Specifically, we say that two intersecting circles C_1 and C_2 intersect at angle θ if at any intersection point of C_1 and C_2 , the angle between the tangent line to C_1 and the tangent line to C_2 equals θ . For any $\theta \in [0, \pi/2]$, the θ -graph $G_\theta(\mathcal{C})$ of a collection of circles \mathcal{C} is the graph with vertex set \mathcal{C} and edges comprising the pairs of circles in \mathcal{C} that intersect at angle θ . In particular, the 0-graph is the tangency graph. We extend Theorem 1 as follows.

► **Theorem 2.** *For every $\theta \in [0, \pi/2]$, there exist θ -graphs of circles in the plane with arbitrarily large girth and chromatic number.*

The proof of Theorem 2 for $\theta > 0$ is significantly simpler than the proof for $\theta = 0$ corresponding to Theorem 1. Therefore, the remainder of the paper is organized as follows. In Section 2, we introduce Gallai’s theorem and prove a version of it with additional constraints as needed for the proof of Theorem 1. In Section 3, we prove Theorem 2 for $\theta > 0$. As the underlying ideas and tools are similar but simpler, this can be considered as a warm-up for the proof of the more involved case $\theta = 0$, which follows in Section 4.

2 Gallai’s theorem with constraints

We start by introducing results from Ramsey theory – Gallai’s theorem and its versions that we need for the proofs of Theorems 1 and 2.

A *homothetic map* in \mathbb{R}^d is a map $h: \mathbb{R}^d \rightarrow \mathbb{R}^d$ of the form $h(p) = p^* + \lambda p$ for some $p^* \in \mathbb{R}^d$ and $\lambda > 0$. In other words, a homothetic map is a composition of (positive) uniform scaling and translation. A set $T' \subseteq \mathbb{R}^d$ is a *homothetic copy* of a set $T \subseteq \mathbb{R}^d$ if there is a homothetic map h in \mathbb{R}^d such that $T' = h(T)$.

The following beautiful theorem, which is a generalization of the well-known van der Waerden’s theorem on arithmetic progressions [21], was first discovered by Gallai in the 1930s, as reported by Rado [18].

33:4 A Solution to Ringel's Circle Problem

► **Gallai's Theorem.** *For every finite set $T \subset \mathbb{R}^d$, there exists a finite set $X \subset \mathbb{R}^d$ such that every k -coloring of X contains a monochromatic homothetic copy of T .*

A *cycle* of length $\ell \geq 2$ on a set X is a tuple (T_1, \dots, T_ℓ) of distinct subsets of X such that there exist distinct elements $x_1, \dots, x_\ell \in X$ with $x_i \in T_i \cap T_{i+1}$ for $i \in [\ell - 1]$ and $x_\ell \in T_\ell \cap T_1$.

In order to guarantee high girth in the proofs of Theorems 1 and 2, we need an appropriate “sparse” version of Gallai's theorem, which excludes short cycles among all homothetic copies of T in X (one of which is guaranteed to be monochromatic). In particular, the following strengthening of Gallai's theorem suffices for the purpose of proving Theorem 2 for $\theta > 0$.

► **Theorem 3** (Prömel, Voigt [17]). *For every finite set $T \subset \mathbb{R}^d$ of size at least 3 and for any integers $g \geq 3$ and $k \geq 1$, there exists a finite set $X \subset \mathbb{R}^d$ such that every k -coloring of X contains a monochromatic homothetic copy of T and no tuple of fewer than g homothetic copies of T in X forms a cycle on X .*

Because Theorem 3 only guarantees the existence of a set X , it is not specific enough to prove Theorem 1. Roughly speaking, in our proof of Theorem 1, we apply a (refined version of) Gallai's theorem to a family of circles in the plane (with $d = 3$, the third coordinate representing the radius) such that the resulting family of circles satisfies a number of additional conditions, e.g., it does not contain two internally tangent circles. To guarantee the additional properties, we develop a refined “sparse” version of Gallai's theorem, which imposes polynomial constraints on the resulting set.

We say that a family \mathcal{F} of $2d$ -variate real polynomials *respects* a set $X \subset \mathbb{R}^d$ if $f(p, q) \neq 0$ for all $f \in \mathcal{F}$ and all pairs of distinct points $p, q \in X$.

► **Theorem 4.** *Let T be a finite subset of \mathbb{R}^d of size at least 3, let \mathcal{F} be a countable family of $2d$ -variate real polynomials that respects T , and let g and k be positive integers. Then there exist a finite set $X \subset \mathbb{R}^d$ and a collection \mathcal{T} of homothetic copies of T in X satisfying the following conditions:*

1. \mathcal{F} respects X ,
2. no tuple of fewer than g homothetic copies of T in \mathcal{T} form a cycle,
3. every k -coloring of X contains a monochromatic homothetic copy of T in \mathcal{T} .

One of the standard ways of proving Gallai's theorem is to derive it from the Hales-Jewett theorem [9]. Our proof of Theorem 4 goes along the same line.

For $m, n \in \mathbb{N}$, a subset L of the n -dimensional m -cube $[m]^n$ is called a *combinatorial line* if there exist a non-empty set of indices $I = \{i_1, \dots, i_k\} \subseteq [n]$ and a choice of $x_i^* \in [m]$ for every $i \in [n] \setminus I$ such that

$$L = \{(x_1, \dots, x_n) \in [m]^n : x_{i_1} = \dots = x_{i_k} \text{ and } x_i = x_i^* \text{ for } i \notin I\}.$$

The indices in I are called the *active coordinates* of L .

► **Hales-Jewett Theorem.** *For any $m, k \in \mathbb{N}$, there exists $n \in \mathbb{N}$ such that every k -coloring of $[m]^n$ contains a monochromatic combinatorial line.*

We need the following “sparse” version of the Hales-Jewett theorem.

► **Theorem 5** (Prömel, Voigt [16]). *For any $m, g, k \in \mathbb{N}$ with $m \geq 3$, there exist $n \in \mathbb{N}$ and a set $H \subseteq [m]^n$ such that every k -coloring of H contains a monochromatic combinatorial line of $[m]^n$ and no tuple of fewer than g combinatorial lines of $[m]^n$ contained in H forms a cycle.*

We also need the following simple algebraic fact.

► **Lemma 6.** For every countable family \mathcal{F} of n -variate real polynomials that are not identically zero, the union of their zero sets $\bigcup_{f \in \mathcal{F}} Z(f)$, where $Z(f) = \{x \in \mathbb{R}^n : f(x) = 0\}$, has empty interior.

Proof. Fix $f \in \mathcal{F}$. Clearly, $Z(f)$ is a closed set in \mathbb{R}^n . Suppose for the sake of contradiction that there is a point x in the interior of $Z(f)$. Let $y \in \mathbb{R}^n$ be such that $f(y) \neq 0$. The univariate polynomial $f_{x,y}$ given by $f_{x,y}(t) = f(x + t(y - x))$ is not identically zero, because $f_{x,y}(1) = f(y) \neq 0$, so it has finitely many roots. However, $f_{x,y}(t) = 0$ whenever $|t|$ is sufficiently small for the point $x + t(y - x)$ to fall into an open neighborhood of x contained in $Z(f)$. There are infinitely many such values t , which is a contradiction. Hence, $Z(f)$ has empty interior. The lemma now follows by the Baire category theorem – a standard tool from topology, which asserts that a countable union of closed sets with empty interior in a complete metric space (such as \mathbb{R}^n with the Euclidean metric) has empty interior. ◀

Now, we are ready to prove Theorem 4.

Proof of Theorem 4. We can assume without loss of generality that \mathcal{F} contains the polynomial δ defined by

$$\delta(p_1, \dots, p_d, q_1, \dots, q_d) = (p_1 - q_1)^2 + \dots + (p_d - q_d)^2,$$

because distinct points $p, q \in \mathbb{R}^d$ satisfy $\delta(p, q) \neq 0$. Put $T = \{t_1, \dots, t_m\}$, where $m = |T|$. Let n and $H \subseteq [m]^n$ be as claimed in Theorem 5 applied to $[m]$, g , and k . For a given vector $\gamma = (\gamma_1, \dots, \gamma_n) \in \mathbb{R}^n$, define a map $\zeta_\gamma: H \rightarrow \mathbb{R}^d$ by $\zeta_\gamma(x) = \sum_{i=1}^n \gamma_i t_{x_i}$, and put $X_\gamma = \zeta_\gamma(H) = \{\zeta_\gamma(x) : x \in H\} \subset \mathbb{R}^d$.

We aim to find a vector $\gamma \in \mathbb{R}^n$ with positive coordinates such that \mathcal{F} respects the set X_γ . For any $f \in \mathcal{F}$ and any distinct $x, y \in H$, let $F_{f,x,y}$ be the n -variate polynomial defined by

$$F_{f,x,y}(\gamma_1, \dots, \gamma_n) = f\left(\sum_{i=1}^n \gamma_i t_{x_i}, \sum_{i=1}^n \gamma_i t_{y_i}\right).$$

Given $f \in \mathcal{F}$ and distinct points $x, y \in H$, let $i \in [n]$ be an index such that $x_i \neq y_i$. Setting $\gamma_i = 1$ and $\gamma_j = 0$ for $j \neq i$, we obtain $F_{f,x,y}(\gamma) = f(t_{x_i}, t_{y_i}) \neq 0$, which shows that $F_{f,x,y}$ is not identically zero. Apply Lemma 6 to the family $\{F_{f,x,y} : f \in \mathcal{F}, x, y \in H, x \neq y\}$ to conclude that the union $\bigcup_{f \in \mathcal{F}, x, y \in H, x \neq y} Z(F_{f,x,y})$ of the zero sets of the polynomials $F_{f,x,y}$ has empty interior. In particular, there exists a vector $\gamma \in \mathbb{R}^n$ with positive coordinates such that $F_{f,x,y}(\gamma) \neq 0$ for all $f \in \mathcal{F}$ and all distinct $x, y \in H$, so that \mathcal{F} respects the set X_γ .

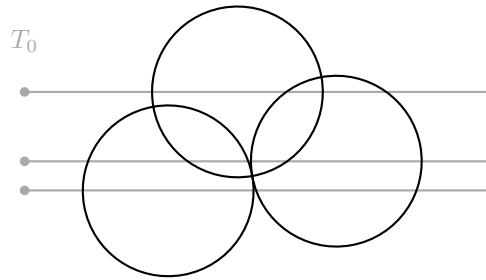
Fix such a vector γ , and let $\zeta = \zeta_\gamma$ and $X = X_\gamma$. Condition 1 thus follows. Furthermore, by our assumption that $\delta \in \mathcal{F}$, we have $\delta(\zeta(x), \zeta(y)) = F_{\delta,x,y}(\gamma) \neq 0$ for any distinct $x, y \in H$, which shows that ζ is injective.

Let \mathcal{L} be the set of combinatorial lines that are contained in H . Every combinatorial line $L \in \mathcal{L}$ gives rise to a homothetic copy of T in X as follows: if I is the set of active coordinates of L and the coordinates $i \notin I$ are fixed to x_i in L , then the set

$$\zeta(L) = \{\zeta(x) : x \in L\} = \left\{ \sum_{i \notin I} \gamma_i t_{x_i} + \left(\sum_{i \in I} \gamma_i \right) t_j : j \in [m] \right\}$$

is a homothetic copy of T . Specifically, we have $\zeta(L) = h(T)$ for the homothetic map h given by $h(p) = p^* + \lambda p$ with $p^* = \sum_{i \notin I} \gamma_i t_{x_i}$ and $\lambda = \sum_{i \in I} \gamma_i > 0$. Let $\mathcal{T} = \{\zeta(L) : L \in \mathcal{L}\}$.

We show that conditions 2 and 3 hold for X and \mathcal{T} . Since no tuple of fewer than g combinatorial lines in \mathcal{L} form a cycle and ζ is injective, no tuple of fewer than g members of \mathcal{T} form a cycle, which is condition 2. For the proof of condition 3, consider a k -coloring ϕ of X . It induces a k -coloring $x \mapsto \phi(\zeta(x))$ of H , in which, by Theorem 5, there is a monochromatic combinatorial line $L \in \mathcal{L}$. We conclude that the homothetic copy $\zeta(L)$ of T in \mathcal{T} is monochromatic in ϕ . ◀



■ **Figure 2** Construction of the set T_0 for $\theta = \pi/2$.

The above proof method can also be used to prove other versions of Gallai's theorem with constraints. On the one hand, we can describe constraints using functions other than polynomials if they satisfy a suitable analogue of Lemma 6, e.g., real analytic functions. On the other hand, we can use other versions of the Hales-Jewett theorem, e.g., the density Hales-Jewett theorem due to Furstenberg and Katznelson [8], which asserts that for any $m \in \mathbb{N}$ and $\alpha > 0$, there is $n \in \mathbb{N}$ such that every subset of $[m]^n$ of size at least αm^n contains a combinatorial line. Then, the same proof leads to the following result.

► **Theorem 7.** *Let T be a finite subset of \mathbb{R}^d , let \mathcal{F} be a countable family of $2d$ -variate real polynomials that respects T , and let $\alpha > 0$. Then there exists a finite set $X \subset \mathbb{R}^d$ such that \mathcal{F} respects X and every subset of X of size at least $\alpha|X|$ contains a homothetic copy of T .*

3 Proof of Theorem 2 for $\theta > 0$

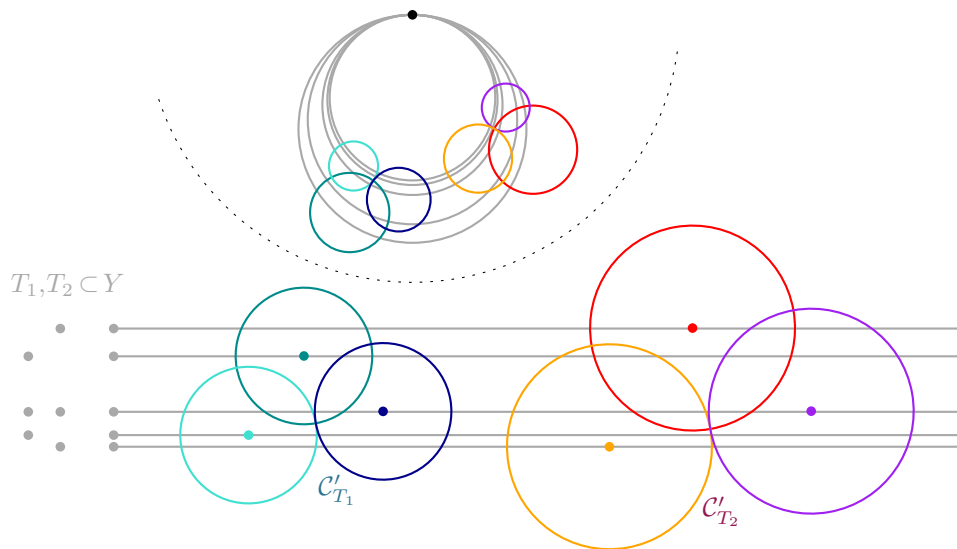
In this section, we prove Theorem 2 for all $\theta \in (0, \pi/2]$. Here is the precise statement.

► **Theorem 8.** *For every $\theta \in (0, \pi/2]$ and any integers $g \geq 3$ and $k \geq 1$, there is a collection of circles \mathcal{C} , no two concentric, such that the θ -graph $G_\theta(\mathcal{C})$ has girth at least g and chromatic number at least k .*

Proof. We fix $\theta \in (0, \pi/2)$ and $g \geq 3$, and construct the families of circles by induction on k . The base case $k \leq 3$ is easy, because all odd cycles can be represented as θ -graphs of circles satisfying the conditions of the theorem.

For the induction step, assume we have already constructed a family of circles \mathcal{C}_k , no two concentric, such that the θ -graph $G_\theta(\mathcal{C})$ has girth at least g and chromatic number at least k , where $k \geq 3$. Let $\mathcal{L}(y)$ denote the horizontal line at coordinate $y \in \mathbb{R}$, that is, $\mathcal{L}(y) = \mathbb{R} \times \{y\}$. To construct a family \mathcal{C}_{k+1} , we perform the following process.

1. *Constructing a "template" set from the family \mathcal{C}_k .* For each circle $C \in \mathcal{C}_k$, we pick all horizontal lines that intersect C at angle θ (meaning that the angle between the horizontal line and the tangent line to C at either of the intersection points is θ), see Figure 2. There are two such lines when $\theta \in (0, \pi/2)$ and only one (through the center of the circle) when $\theta = \pi/2$. By slightly rotating the family \mathcal{C} if needed, we can guarantee that these lines are all distinct. When $\theta = \pi/2$, this requires the additional assumption that no two circles in \mathcal{C} are concentric (which is otherwise superfluous). Let $T_0 \subset \mathbb{R}$ be the set of y -coordinates of the lines.
2. *Applying Gallai's theorem.* Theorem 3 in \mathbb{R} applied to the set T_0 yields a finite set $Y \subset \mathbb{R}$ such that every k -coloring of Y contains a monochromatic homothetic copy of T_0 and no tuple of fewer than $\lceil g/2 \rceil$ homothetic copies of T_0 in Y form a cycle.



■ **Figure 3** Illustration for steps 3–5 in the proof of Theorem 8 for $\theta = \pi/2$: Construction of a preliminary family from Y and the result after inversion with respect to the dotted circle. Note that the set Y consists of two homothetic copies of the set T_0 from Fig. 2 that have one element in common.

3. *Geometric interpretation of the resulting set.* The set Y gives rise to the family of horizontal lines $\mathcal{L}' = \{\mathcal{L}(y) : y \in Y\}$. Let \mathcal{T} be the family of homothetic copies of T_0 in Y .
4. *Attaching a copy of \mathcal{C}_k for each homothetic copy of the “template”.* For each $T \in \mathcal{T}$, we consider the set of horizontal lines $\mathcal{L}'_T = \{\mathcal{L}(y) : y \in T\}$ and construct a homothetic copy \mathcal{C}'_T of \mathcal{C}_k such that each line in \mathcal{L}'_T intersects a single circle \mathcal{C}'_T at angle θ ; see Figure 3. (Each circle in \mathcal{C}'_T intersects two lines in \mathcal{L}'_T at angle θ when $\theta \in (0, \pi/2)$ and only one when $\theta = \pi/2$.) This is possible because the set of lines $\{\mathcal{L}(y) : y \in T_0\}$ has this property with respect to \mathcal{C}_k , and T is a homothetic copy of T_0 . We spread the copies \mathcal{C}'_T horizontally so that a vertical line separates \mathcal{C}'_{T_1} from \mathcal{C}'_{T_2} for any distinct $T_1, T_2 \in \mathcal{T}$. Let $\mathcal{C}' = \bigcup_{T \in \mathcal{T}} \mathcal{C}'_T$.
5. *Constructing the final family \mathcal{C}_{k+1} via inversion.* Finally, we construct the family \mathcal{C}_{k+1} by applying a geometric inversion to the lines and circles in $\mathcal{L}' \cup \mathcal{C}'$, where the center of inversion is chosen not to lie on any of these lines or circles. See Figure 3 for an illustration. By basic properties of inversion, the resulting family consists only of circles; in particular, the lines in \mathcal{L}' turn into a bunch of circles tangent to the horizontal line at the center of inversion [3, chapter 6]. To ensure that the inversion does not create concentric circles, we choose the center of inversion not to lie on any line passing through the centers of two circles in \mathcal{C}' or any vertical line passing through the center of a circle in \mathcal{C}' .

We claim that the θ -graph $G_\theta(\mathcal{C}_{k+1})$ has girth at least g and chromatic number at least $k+1$. Since inversion preserves angles, this graph is isomorphic to the θ -graph of $\mathcal{L}' \cup \mathcal{C}'$ (which is defined analogously to the θ -graph for a collection of circles). Let G denote the latter θ -graph. It is thus sufficient to prove that G has girth at least g and chromatic number at least $k+1$.

To this end, we observe that by the construction, G has the following structure: for every $T \in \mathcal{T}$, the subgraph of G induced on the vertices in \mathcal{C}'_T is isomorphic to $G_\theta(\mathcal{C}_k)$, and the remaining edges form a collection of bipartite subgraphs between the vertices in \mathcal{C}'_T and the vertices in \mathcal{L}'_T , where each vertex in \mathcal{C}'_T is adjacent to two corresponding vertices in \mathcal{L}'_T if $\theta \in (0, \pi/2)$ and only one if $\theta = \pi/2$.

We exploit the structure above in the proofs of the final two claims. They are standard when applying generalizations of Tutte’s construction; see, e.g., [4, 14].

▷ Claim 8.1. The graph G has girth at least g .

Proof. For every $T \in \mathcal{T}$, every cycle in G that lies entirely within \mathcal{C}'_T has length at least g because the subgraph of G induced on the vertices in \mathcal{C}'_T is isomorphic to $G_\theta(\mathcal{C}_k)$, the girth of which is at least g by the induction hypothesis. It thus remains to consider a cycle in G of length $\ell \geq 3$ that does not lie entirely within \mathcal{C}'_T for any $T \in \mathcal{T}$. It must contain vertices from \mathcal{L}' , say, L_1, \dots, L_m in this order along the cycle. For each $i \in [m]$, since L_i has no edges to the rest of \mathcal{L}' and at most one edge to \mathcal{C}'_T for each $T \in \mathcal{T}$, the neighbors of L_i on the cycle lie in two different sets of the form \mathcal{C}'_T . For each $i \in [m]$, let $T_i \in \mathcal{T}$ be such that the part of the cycle between L_i and L_{i+1} (or L_1 when $i = m$) lies within \mathcal{C}'_{T_i} .

It follows that (T_1, \dots, T_m) is a cycle in \mathcal{T} of length m or contains such a cycle if some members of \mathcal{T} repeat among T_1, \dots, T_m . Hence, Theorem 3 yields $m \geq \lceil g/2 \rceil$. Since the cycle contains at least one vertex from \mathcal{C}' between L_i and L_{i+1} (or L_1 when $i = m$) for any $i \in [m]$, we conclude that $\ell \geq 2m \geq g$. ◁

▷ Claim 8.2. The graph G has chromatic number at least $k + 1$.

Proof. Suppose for the sake of contradiction that the graph G is k -colorable. Pick a proper k -coloring of G , and consider its restriction to the vertices in \mathcal{L}' . It induces a k -coloring of Y via the correspondence $Y \ni y \leftrightarrow \mathfrak{L}(y) \in \mathcal{L}'$. It follows from the application of Theorem 3 that there is a monochromatic homothetic copy T of T_0 in \mathcal{T} , which means that the set of lines \mathcal{L}'_T is monochromatic. Since the edges of G that connect these lines with \mathcal{C}'_T match all of \mathcal{C}'_T , their common color does not occur on the circles in \mathcal{C}'_T . Therefore, the given k -coloring of G induces a proper $(k - 1)$ -coloring of the subgraph of G induced on the vertices in \mathcal{C}'_T , which is isomorphic to $G_\theta(\mathcal{C}_k)$. This contradicts the assumption that the graph $G_\theta(\mathcal{C}_k)$ has chromatic number at least k . ◁

This completes the proof of Theorem 8 by induction. ◀

Observe that the proof above cannot be used for $\theta = 0$ to prove Theorem 1, because the inversion at step 5 turns all lines in \mathcal{L}' into circles tangent at one point, so the resulting collection of circles is not a constellation (and the resulting tangency graph has girth 3).

4 Proof of Theorem 1

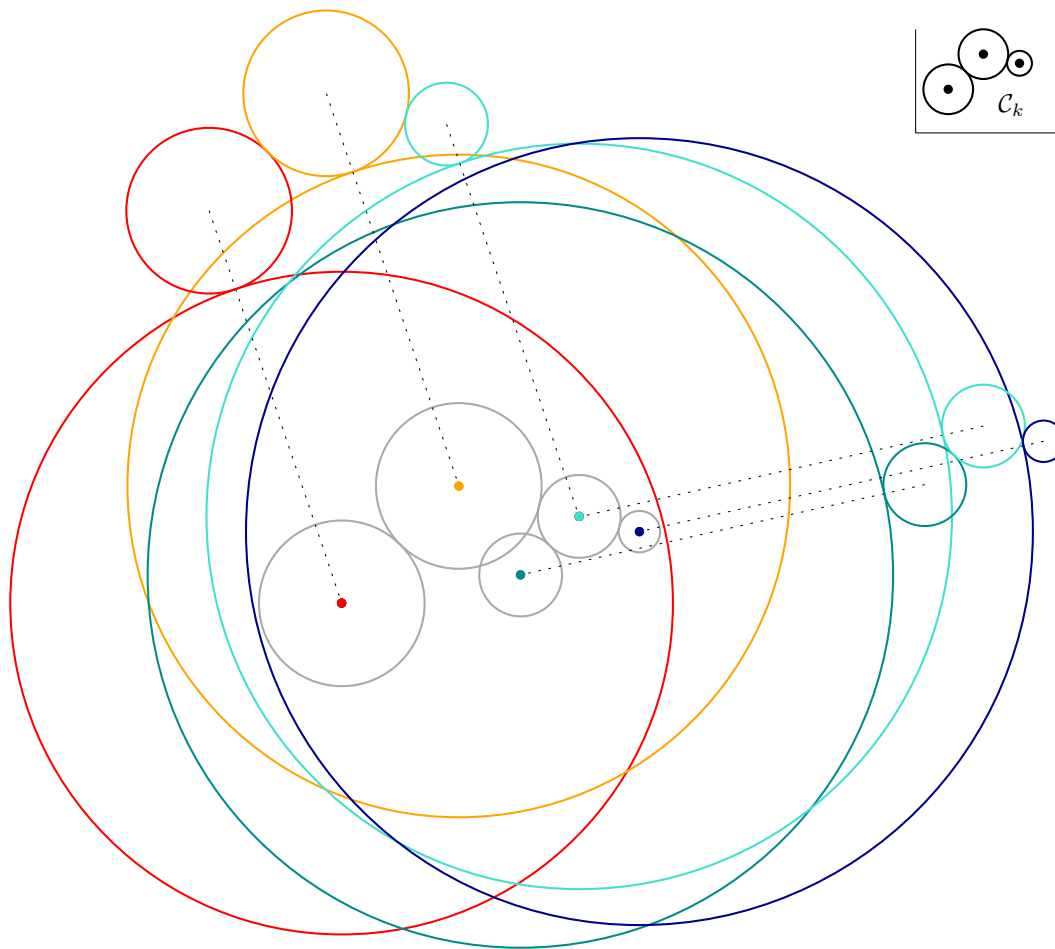
In this section, we prove Theorem 1. For the purpose of induction, we prove the following stronger statement, which directly implies Theorem 1.

► **Theorem 9.** *For any integers $g \geq 3$ and $k \geq 1$, there exists a collection of circles \mathcal{C} , no two concentric and no two internally tangent, such that the tangency graph $G(\mathcal{C})$ has girth at least g and chromatic number at least k .*

For the sake of clarity, we first present the construction of the families \mathcal{C} and then we prove that the construction satisfies the requirements of the theorem.

4.1 High-level description of the construction

We fix $g \geq 3$ and prove the theorem by induction on k . The base case $k \leq 3$ is easy, because all odd cycles can be represented as tangency graphs of circles satisfying the conditions of the theorem.



■ **Figure 4** Illustration for the construction of the family \mathcal{C}_{k+1} from \mathcal{C}_k . Gray circles represent a part of the set $X \subset \mathbb{R}^3$ containing two homothetic copies of T_0 in \mathcal{T} . The family \mathcal{C}_{k+1} contains a (large) circle for each point in X and a homothetic copy of \mathcal{C}_k for each homothetic copy of T_0 in \mathcal{T} .

For the induction step, assume we have already constructed a family of circles \mathcal{C}_k , no two concentric and no two internally tangent, such that the tangency graph $G(\mathcal{C}_k)$ has girth at least g and chromatic number at least k . Let $\mathfrak{C}(x, y, r)$ denote the circle with center $(x, y) \in \mathbb{R}^2$ and radius $r > 0$. To construct a family \mathcal{C}_{k+1} with girth g and chromatic number $k + 1$, we perform the following process. See Figure 4 for an illustration.

1. *Constructing a “template” set from the family \mathcal{C}_k .* We represent each circle $\mathfrak{C}(x, y, r) \in \mathcal{C}_k$ by the point $(x, y, r) \in \mathbb{R}^3$, to obtain the set

$$T_0 = \{(x, y, r) \in \mathbb{R}^3 : \mathfrak{C}(x, y, r) \in \mathcal{C}_k\} \subset \mathbb{R}^3.$$

2. *Applying Gallai’s theorem with constraints.* Theorem 4 in \mathbb{R}^3 applied to the set T_0 with appropriate constraints to be detailed below yields a finite set $X \subset \mathbb{R}^3$ and a collection \mathcal{T} of homothetic copies of T_0 in X such that every k -coloring of X contains a monochromatic homothetic copy of T_0 in \mathcal{T} and no tuple of fewer than $\lceil g/3 \rceil$ homothetic copies of T_0 in \mathcal{T} form a cycle. For each $T \in \mathcal{T}$, let h_T be the homothetic map from T_0 to T in \mathbb{R}^3 . It has the following form for some $x_T^*, y_T^*, r_T^* \in \mathbb{R}$ and $\lambda_T > 0$:

$$h_T: \mathbb{R}^3 \ni (x, y, r) \mapsto (x_T^* + \lambda_T x, y_T^* + \lambda_T y, r_T^* + \lambda_T r) \in \mathbb{R}^3.$$

33:10 A Solution to Ringel's Circle Problem

3. *Geometric interpretation of the resulting set.* Let $R_0 = \max\{r' : (x', y', r') \in X\}$, and let $R \in \mathbb{R}$ satisfy $R > R_0$. (In the sequel, R is going to be “large”.) The set X gives rise to a family of “large” circles \mathcal{C}'_R , parameterized by R , defined as follows:

$$\mathcal{C}'_R = \{\mathfrak{C}(x', y', R - r') : (x', y', r') \in X\}.$$

The use of the stronger Theorem 4 with appropriate constraints, rather than Theorem 3, allows us to infer that no two circles in \mathcal{C}'_R are concentric or internally tangent.

4. *Attaching a copy of \mathcal{C}_k for each homothetic copy of the “template”.* We pick a set $\{\phi_T\}_{T \in \mathcal{T}}$ of distinct angles in $[0, \pi)$ that satisfy a certain condition to be detailed below. For every $T \in \mathcal{T}$ and every circle $C = \mathfrak{C}(x, y, r) \in \mathcal{C}_k$, we define the following two circles, where $(x', y', r') = h_T(x, y, r) \in T$:

$$\begin{aligned} \mu_{R,T}(C) &= \mathfrak{C}(x', y', R - r'), \quad \text{which is a circle in } \mathcal{C}'_R, \\ \nu_{R,T}(C) &= \mathfrak{C}(x' + (R - r'_T) \cos(\phi_T), y' + (R - r'_T) \sin(\phi_T), \lambda_T r). \end{aligned}$$

In words, $\mu_{R,T}(C)$ is a “large” circle with center (x', y') , and $\nu_{R,T}(C)$ is a “small” circle with center translated from (x', y') in direction ϕ_T , externally tangent to $\mu_{R,T}(C)$. For every $T \in \mathcal{T}$, we set

$$\mathcal{C}'_{R,T} = \{\nu_{R,T}(C) : C \in \mathcal{C}_k\}.$$

Since the angles ϕ_T are distinct and the radii of the circles $\nu_{R,T}(C)$ do not depend on R , when R is sufficiently large, the circles \mathcal{C}'_{R,T_1} are disjoint from those in \mathcal{C}'_{R,T_2} for any distinct $T_1, T_2 \in \mathcal{T}$.

5. *Constructing the final family \mathcal{C}_{k+1} .* Finally, we define

$$\mathcal{C}''_R = \mathcal{C}'_R \cup \bigcup_{T \in \mathcal{T}} \mathcal{C}'_{R,T}.$$

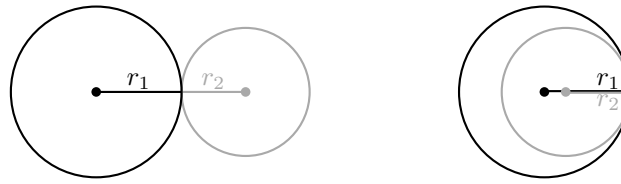
We will show that $\mathcal{C}_{k+1} := \mathcal{C}''_R$ satisfies all claimed properties if R is sufficiently large.

Comparison with the construction in Section 3

This construction follows the general strategy of the construction for the θ -graph presented in Section 3. Notable differences result from the need to avoid multiple circles mutually tangent at one point, which arise in the last step of that construction when applying inversion.

1. While in Section 3, we have $T_0 \subset \mathbb{R}$, here we have to resort to the more complex choice of $T_0 \subset \mathbb{R}^3$.
2. While in Section 3, the standard “sparse” version of Gallai’s theorem is sufficient, here we need the stronger version with constraints, to be able to infer that the resulting set of circles avoids concentricities and internal tangencies.
3. The construction of “large” circles here is explicit and uses a parameter R that must be chosen appropriately, while in Section 3, horizontal lines are used instead.
4. The construction of “small” circles here is more involved and uses a set of parameters $\{\phi_T\}_{T \in \mathcal{T}}$ that must be chosen appropriately.
5. In Section 3, a final application of inversion is required to transform the horizontal lines into circles, which is no longer needed in the construction here.

The proof of validity of the construction is somewhat more complex, accordingly.



■ **Figure 5** External and internal tangency of circles.

4.2 Proof of Theorem 9

In the proof, we use the following two simple lemmas in addition to Theorem 4.

- ▶ **Lemma 10.** *Circles $\mathcal{C}(x_1, y_1, r_1)$ and $\mathcal{C}(x_2, y_2, r_2)$ are*
 - *externally tangent if and only if $(x_1 - x_2)^2 + (y_1 - y_2)^2 = (r_1 + r_2)^2$,*
 - *internally tangent if and only if $(x_1 - x_2)^2 + (y_1 - y_2)^2 = (r_1 - r_2)^2$.*

Proof. Circles $\mathcal{C}(x_1, y_1, r_1)$ and $\mathcal{C}(x_2, y_2, r_2)$ are externally tangent if and only if the segment connecting their centers (x_1, x_2) and (y_1, y_2) has length $r_1 + r_2$, and they are internally tangent if and only if it has length $|r_1 - r_2|$. See Figure 5 for an illustration. ◀

- ▶ **Lemma 11.** *If $a, b, c, \varphi \in \mathbb{R}$, $(a, b) \neq (0, 0)$, and the vectors (a, b) and $(\cos \varphi, \sin \varphi)$ are not parallel in \mathbb{R}^2 , then the equality $(a + R \cos \varphi)^2 + (b + R \sin \varphi)^2 = (c + R)^2$ holds for at most one value $R \in \mathbb{R}$.*

Proof. Consider the univariate polynomial f defined by

$$\begin{aligned} f(R) &= (a + R \cos \varphi)^2 + (b + R \sin \varphi)^2 - (c + R)^2 \\ &= (a^2 + b^2 - c^2) + 2R(a \cos \varphi + b \sin \varphi - c). \end{aligned}$$

It is identically zero only if $a^2 + b^2 = c^2$ and $a \cos \varphi + b \sin \varphi = c$. However, if $a^2 + b^2 = c^2$ and the vectors (a, b) and $(\cos \varphi, \sin \varphi)$ are not parallel, then the Cauchy-Schwarz inequality yields $|a \cos \varphi + b \sin \varphi| < \sqrt{a^2 + b^2} \cdot \sqrt{\cos^2 \varphi + \sin^2 \varphi} = |c|$, so $a \cos \varphi + b \sin \varphi \neq c$. Since f is not identically zero and has degree at most 1, it has at most one root in \mathbb{R} . ◀

We are now ready to present the details of the proof of Theorem 9.

Proof of Theorem 9. We proceed by induction on k and, for the induction step, construct the family of circles \mathcal{C}_{k+1} from a family of circles \mathcal{C}_k as was described above. The following claim describes the property of the set X constructed by applying our enhanced version of Gallai’s theorem, namely Theorem 4, with appropriate polynomial constraints to T_0 .

▷ **Claim 9.1.** There exists a finite set $X \subset \mathbb{R}^3$ and a collection \mathcal{T} of homothetic copies of T_0 in X with the following properties:

1. for any two distinct points $(x_1, y_1, r_1), (x_2, y_2, r_2) \in X$, we have
 - a. $(x_1, y_1) \neq (x_2, y_2)$,
 - b. $(x_1 - x_2)^2 + (y_1 - y_2)^2 \neq (r_1 - r_2)^2$,
2. no tuple of fewer than $\lceil g/3 \rceil$ homothetic copies of T_0 in \mathcal{T} form a cycle,
3. every k -coloring of X contains a monochromatic homothetic copy of T_0 in \mathcal{T} .

Proof. Consider the following two 6-variate polynomials:

$$\begin{aligned} f_a(x_1, y_1, r_1, x_2, y_2, r_2) &= (x_1 - x_2)^2 + (y_1 - y_2)^2, \\ f_b(x_1, y_1, r_1, x_2, y_2, r_2) &= (x_1 - x_2)^2 + (y_1 - y_2)^2 - (r_1 - r_2)^2. \end{aligned}$$

33:12 A Solution to Ringel's Circle Problem

We have $f_a(x_1, y_1, r_1, x_2, y_2, r_2) \neq 0$ if and only if $(x_1, y_1) \neq (x_2, y_2)$, which holds in particular for distinct points $(x_1, y_1, r_1), (x_2, y_2, r_2) \in T_0$, by the assumption that no two circles in \mathcal{C}_k are concentric. Lemma 10 and the assumption that no two circles in \mathcal{C} are internally tangent imply $f_b(x_1, y_1, r_1, x_2, y_2, r_2) \neq 0$ for any distinct points $(x_1, y_1, r_1), (x_2, y_2, r_2) \in T_0$. Theorem 4 applied to T_0 , $\mathcal{F} = \{f_a, f_b\}$, $\lceil g/3 \rceil$, and k directly yields the requested set X and collection \mathcal{T} . \triangleleft

For the construction of the families of circles $\{\nu_{R,T}\}_{T \in \mathcal{T}}$, we let $\{\phi_T\}_{T \in \mathcal{T}}$ be a set of distinct angles in $[0, \pi)$ such that for every $T \in \mathcal{T}$, the unit vector $(\cos \phi_T, \sin \phi_T) \in \mathbb{R}^2$ is not parallel to the vector $(x_1 - x_2, y_1 - y_2)$ for any distinct points $(x_1, y_1, r_1), (x_2, y_2, r_2) \in X$, where the latter vector is non-zero, by condition 1a of Claim 9.1.

We now claim that for a sufficiently large R , the family \mathcal{C}''_R defined above satisfies the following conditions on concentricity and tangency.

▷ **Claim 9.2.** The following holds when R is sufficiently large:

1. no two circles in \mathcal{C}''_R are concentric,
2. no two circles in \mathcal{C}''_R are internally tangent,
3. a pair of circles in \mathcal{C}''_R is externally tangent if and only if it belongs to one of the two following types:
 - a. $\mu_{R,T}(C)$ and $\nu_{R,T}(C)$ for any $T \in \mathcal{T}$ and any $C \in \mathcal{C}_k$,
 - b. $\nu_{R,T}(C_1)$ and $\nu_{R,T}(C_2)$ for any $T \in \mathcal{T}$ and any $C_1, C_2 \in \mathcal{C}_k$ that are externally tangent.

Proof. First, consider two distinct circles $C' = \mathfrak{C}(x', y', R - r')$ and $C'' = \mathfrak{C}(x'', y'', R - r'')$ in \mathcal{C}''_R , where $(x', y', r'), (x'', y'', r'') \in X$. By condition 1a of Claim 9.1, the circles C' and C'' are not concentric. By Lemma 10, the circles C' and C'' are internally tangent if and only if $(x' - x'')^2 + (y' - y'')^2 = (r' - r'')^2$, which does not hold due to condition 1b of Claim 9.1. Also by Lemma 10, the circles C' and C'' are externally tangent if and only if $(x' - x'')^2 + (y' - y'')^2 = (2R - r' - r'')^2$, which does not hold when R is sufficiently large.

Next, let $T \in \mathcal{T}$, and consider two distinct circles C'_1 and C'_2 such that for $i \in [2]$, we have $C'_i = \mathfrak{C}(x'_i + (R - r_T^*) \cos \phi_T, y'_i + (R - r_T^*) \sin \phi_T, \lambda_T r_i) = \nu_{R,T}(C_i)$, where $C_i = \mathfrak{C}(x_i, y_i, r_i) \in \mathcal{C}_k$ and $h_T(x_i, y_i, r_i) = (x'_i, y'_i, r'_i) \in T$. The assumption that the circles C_1 and C_2 are not concentric, that is, $(x_1, y_1) \neq (x_2, y_2)$, yields $(x'_1, y'_1) \neq (x'_2, y'_2)$, which implies that the circles C'_1 and C'_2 are not concentric. By Lemma 10, the circles C'_1 and C'_2 are internally tangent if and only if $(x'_1 - x'_2)^2 + (y'_1 - y'_2)^2 = \lambda_T^2 (r_1 - r_2)^2$, which is equivalent to $(x_1 - x_2)^2 + (y_1 - y_2)^2 = (r_1 - r_2)^2$, which does not hold due to the assumption that C_1 and C_2 are not internally tangent. Also by Lemma 10, the circles C'_1 and C'_2 are externally tangent if and only if $(x'_1 - x'_2)^2 + (y'_1 - y'_2)^2 = \lambda_T^2 (r_1 + r_2)^2$, which is equivalent to $(x_1 - x_2)^2 + (y_1 - y_2)^2 = (r_1 + r_2)^2$, which means that C_1 and C_2 are externally tangent.

Next, for two distinct $T_1, T_2 \in \mathcal{T}$, consider circles of the form $C'_1 = \nu_{R,T_1}(C_1)$ and $C'_2 = \nu_{R,T_2}(C_2)$, where $C_1, C_2 \in \mathcal{C}_k$. Since $\phi_{T_1} \neq \phi_{T_2}$ and the radii of C'_1 and C'_2 are independent of R , the circles C'_1 and C'_2 are arbitrarily far apart as R grows. In particular, they are disjoint and not nested when R is sufficiently large.

Finally, consider a circle $C' = \mathfrak{C}(x', y', R - r') \in \mathcal{C}''_R$, where $(x', y', r') \in X$, and a circle $C'' = \mathfrak{C}(x'' + (R - r_T^*) \cos \phi_T, y'' + (R - r_T^*) \sin \phi_T, \lambda_T r) = \nu_{R,T}(C)$, where $C = \mathfrak{C}(x, y, r) \in \mathcal{C}_k$ and $h_T(x, y, r) = (x'', y'', r'') \in T$. When $R > r_T^*$, since the vectors $(x'' - x', y'' - y')$ and $(\cos \phi_T, \sin \phi_T)$ are not parallel unless $(x', y') = (x'', y'')$, the circles C' and C'' are not concentric. By Lemma 10, the circles C' and C'' are externally tangent if and only if

$$(x'' - x' + (R - r_T^*) \cos \phi_T)^2 + (y'' - y' + (R - r_T^*) \sin \phi_T)^2 = (R - r' + \lambda_T r)^2,$$

which is equivalent to

$$((x'' - x') + R' \cos \phi_T)^2 + ((y'' - y') + R' \sin \phi_T)^2 = ((r'' - r') + R')^2,$$

where $R' = R - r_T^*$. By Lemma 11, the equality above holds for at most one value of R' (so at most one value of R) unless $(x', y', r') = (x'', y'', r'')$, in which case it clearly holds for every value of R' (so every value of R). The latter case means that $C''' = \mu_{R,T}(C)$, as requested in case 3a. Also by Lemma 10, the circles C' and C''' are internally tangent if and only if

$$(x'' - x' + (R - r_T^*) \cos \phi_T)^2 + (y'' - y' + (R - r_T^*) \sin \phi_T)^2 = (R - r' - \lambda_T r)^2,$$

which is equivalent to

$$((x'' - x') + R' \cos \phi_T)^2 + ((y'' - y') + R' \sin \phi_T)^2 = ((2r_T^* - r' - r'') + R')^2,$$

where $R' = R - r_T^*$. By Lemma 11, the equality above holds for at most one value of R' (so at most one value of R) unless $(x', y', r') = (x'', y'', r'')$. In the latter case, C' and C''' are externally tangent (as we have shown previously), so they cannot be internally tangent. \triangleleft

Let $R > R_0$ be sufficiently large for the conclusions of Claim 9.2 to hold. Conditions 2 and 3 of Claim 9.2 imply the following structure of the tangency graph $G(\mathcal{C}_R'')$: for every $T \in \mathcal{T}$, the subgraph induced on the vertices in $\mathcal{C}'_{R,T}$ is isomorphic to $G(\mathcal{C}_k)$ and the remaining edges form a collection of matchings between the vertices in $\mathcal{C}'_{R,T}$ (which are of the form $\nu_{R,T}(C)$ for $C \in \mathcal{C}_k$) and the vertices in \mathcal{C}'_R of the form $\mu_{R,T}(C)$ for $C \in \mathcal{C}_k$. We exploit this structure in the proofs of the final two claims, which are analogous to Claims 8.1 and 8.2.

\triangleright Claim 9.3. The tangency graph $G(\mathcal{C}_R'')$ has girth at least g .

Proof. Let $G = G(\mathcal{C}_R'')$. For every $T \in \mathcal{T}$, since the subgraph of G induced on the vertices in $\mathcal{C}'_{R,T}$ is isomorphic to $G(\mathcal{C}_k)$, the girth of which is at least g by the induction hypothesis, every cycle in G that lies entirely within $\mathcal{C}'_{R,T}$ has length at least g . Consider now a cycle in G of length $\ell \geq 3$ that does not lie entirely within $\mathcal{C}'_{R,T}$ for any $T \in \mathcal{T}$. It must contain vertices from \mathcal{C}'_R , say, C_1, \dots, C_m in this order along the cycle. For each $i \in [m]$, since C_i has no edges to the rest of \mathcal{C}'_R and at most one edge to $\mathcal{C}'_{R,T}$ for each $T \in \mathcal{T}$, the neighbors of C_i on the cycle lie in two different sets of the form $\mathcal{C}'_{R,T}$. For each $i \in [m]$, let $T_i \in \mathcal{T}$ be such that the part of the cycle between C_i and C_{i+1} (or C_1 if $i = m$) lies within \mathcal{C}'_{R,T_i} . It follows that (T_1, \dots, T_m) is a cycle in \mathcal{T} of length m or contains such a cycle if some members of \mathcal{T} repeat among T_1, \dots, T_m . Condition 2 of Claim 9.1 yields $m \geq \lceil g/3 \rceil$. Since there are at least two vertices from \mathcal{C}'_{R,T_i} between C_i and C_{i+1} (or C_1 when $i = m$) for any $i \in [m]$, we conclude that $\ell \geq 3m \geq g$. \triangleleft

\triangleright Claim 9.4. The tangency graph $G(\mathcal{C}_R'')$ has chromatic number at least $k + 1$.

Proof. Suppose for the sake of contradiction that the graph $G = G(\mathcal{C}_R'')$ is k -colorable. Pick a proper k -coloring of G , and consider its restriction to the vertices in \mathcal{C}'_R . It induces a k -coloring of X via the correspondence $X \ni (x', y', r') \leftrightarrow \mathfrak{C}(x', y', R - r') \in \mathcal{C}'_R$. By condition 3 of Claim 9.1, there is a monochromatic homothetic copy T of T_0 in \mathcal{T} , which means that the set of circles $\{\mu_{R,T}(C) : C \in \mathcal{C}_k\}$ is monochromatic. Since these circles are connected to $\mathcal{C}'_{R,T}$ by a perfect matching in G , their common color does not occur on the circles in $\mathcal{C}'_{R,T}$. Therefore, the given k -coloring of G induces a proper $(k - 1)$ -coloring of the graph $G(\mathcal{C}'_{R,T})$, which is isomorphic to $G(\mathcal{C}_k)$. This contradicts the assumption that the graph $G(\mathcal{C}_k)$ has chromatic number at least k . \triangleleft

We complete the proof of the induction step by setting $\mathcal{C}_{k+1}'' = \mathcal{C}_R''$ and observing that the induction statement follows from Claims 9.2 (conditions 1 and 2), 9.3, and 9.4. \blacktriangleleft

References

- 1 Kenneth Appel and Wolfgang Haken. Every planar map is four colorable. Part I: Discharging. *Illinois Journal of Mathematics*, 21(3):429–490, 1977.
- 2 Kenneth Appel, Wolfgang Haken, and John Koch. Every planar map is four colorable. Part II: Reducibility. *Illinois Journal of Mathematics*, 21(3):491–567, 1977.
- 3 Harold Scott Macdonald Coxeter. *Introduction to geometry*. John Wiley & Sons, 1961.
- 4 James Davies. Box and segment intersection graphs with large girth and chromatic number. *Advances in Combinatorics*, 2021:7, 9 pp., 2021. doi:10.19086/aic.25431.
- 5 Aubrey D. N. J. de Grey. The chromatic number of the plane is at least 5. *Geombinatorics*, 28:5–18, 2018.
- 6 Blanche Descartes. A three colour problem. *Eureka*, 9(21):24–25, 1947.
- 7 Blanche Descartes. Solution to advanced problem no. 4526. *The American Mathematical Monthly*, 61:352, 1954.
- 8 Hillel Furstenberg and Yitzhak Katznelson. A density version of the Hales-Jewett theorem. *Journal d'Analyse Mathématique*, 57:64–119, 1991. doi:10.1007/BF03041066.
- 9 Andrew W. Hales and Robert I. Jewett. Regularity and positional games. *Transactions of the American Mathematical Society*, 106(2):222–229, 1963.
- 10 Brad Jackson and Gerhard Ringel. Colorings of circles. *The American Mathematical Monthly*, 91(1):42–49, 1984. doi:10.1080/00029890.1984.11971333.
- 11 Tommy R. Jensen and Bjarne Toft. *Graph Coloring Problems*. John Wiley & Sons, 1995.
- 12 Gil Kalai. Some old and new problems in combinatorial geometry I: Around Borsuk's problem. In Artur Czumaj, Agelos Georgakopoulos, Daniel Král, Vadim Lozin, and Oleg Pikhurko, editors, *Surveys in Combinatorics*, volume 424 of *London Mathematical Society Lecture Note Series*, pages 147–174. Cambridge University Press, 2015.
- 13 Paul Koebe. Kontaktprobleme der konformen Abbildung. *Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physische Klasse*, 88:141–164, 1936.
- 14 Alexandr V. Kostochka and Jaroslav Nešetřil. Properties of Descartes' construction of triangle-free graphs with high chromatic number. *Combinatorics, Probability and Computing*, 8(5):467–472, 1999. doi:10.1017/S0963548399004022.
- 15 János Pach. Finite point configurations. In Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, pages 26–50. CRC Press, 3rd edition, 2017.
- 16 Hans Jürgen Prömel and Bernd Voigt. A sparse Graham-Rothschild theorem. *Transactions of the American Mathematical Society*, 309(1):113–137, 1988. doi:10.1090/S0002-9947-1988-0957064-5.
- 17 Hans Jürgen Prömel and Bernd Voigt. A sparse Gallai-Witt theorem. In Rainer Bodendiek and Rudolf Henn, editors, *Topics in Combinatorics and Graph Theory*, pages 747–755. Physica-Verlag Heidelberg, 1990. doi:10.1007/978-3-642-46908-4_84.
- 18 Richard Rado. Note on combinatorial analysis. *Proceedings of the London Mathematical Society*, 48:122–160, 1945.
- 19 Gerhard Ringel. *Färbungsprobleme auf Flächen und Graphen*, volume 2 of *Mathematische Monographien*. VEB Deutscher Verlag der Wissenschaften, 1959.
- 20 Alexander Soifer. *The Mathematical Coloring Book: Mathematics of Coloring and the Colorful Life of its Creators*. Springer, 2009.
- 21 Bartel L. van der Waerden. Beweis einer Baudetschen Vermutung. *Nieuw Archief voor Wiskunde*, 15:212–216, 1927.

Computing Generalized Rank Invariant for 2-Parameter Persistence Modules via Zigzag Persistence and Its Applications

Tamal K. Dey ✉

Department of Computer Science, Purdue University, West Lafayette, IN, USA

Woojin Kim ✉

Department of Mathematics, Duke University, Durham, NC, USA

Facundo Mémoli ✉

Department of Mathematics and Department of Computer Science and Engineering,
The Ohio State University, Columbus, OH, USA

Abstract

The notion of generalized rank invariant in the context of multiparameter persistence has become an important ingredient for defining interesting homological structures such as generalized persistence diagrams. Naturally, computing these rank invariants efficiently is a prelude to computing any of these derived structures efficiently. We show that the generalized rank over a finite interval I of a \mathbf{Z}^2 -indexed persistence module M is equal to the generalized rank of the zigzag module that is induced on a certain path in I tracing mostly its boundary. Hence, we can compute the generalized rank over I by computing the barcode of the zigzag module obtained by restricting the bifiltration inducing M to that path. If the bifiltration and I have at most t simplices and points respectively, this computation takes $O(t^\omega)$ time where $\omega \in [2, 2.373)$ is the exponent of matrix multiplication. Among others, we apply this result to obtain an improved algorithm for the following problem. Given a bifiltration inducing a module M , determine whether M is interval decomposable and, if so, compute all intervals supporting its summands.

2012 ACM Subject Classification Mathematics of computing \rightarrow Topology; Theory of computation \rightarrow Computational geometry

Keywords and phrases Multiparameter persistent homology, Zigzag persistent homology, Generalized Persistence Diagrams, Möbius inversion

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.34

Related Version *Full Version*: <https://arxiv.org/abs/2111.15058>

Funding This work is supported by NSF grants CCF-2049010, CCF-1740761, DMS-1547357, and IIS-1901360.

Acknowledgements The authors thank the anonymous reviewers for constructive feedback and suggesting ideas that shortened the proof of Theorem 24.

1 Introduction

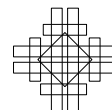
In Topological Data Analysis (TDA) one of the central tasks is that of decomposing persistence modules into direct sums of *indecomposables*. In the case of a persistence module M over the integers \mathbf{Z} , the indecomposables are interval modules, which implies that M is isomorphic to a direct sum of *interval* modules $\mathbb{I}([b_\alpha, d_\alpha])$, for integers $b_\alpha \leq d_\alpha$ and α in some index set A . This follows from a classification theorem for quiver representations established by Pierre Gabriel in the 1970s. The multiset of intervals $\{[b_\alpha, d_\alpha], \alpha \in A\}$ that appear in this decomposition constitutes the *persistence diagram*, or equivalently, the *barcode* of M – a central object in TDA [19, 21].



© Tamal K. Dey, Woojin Kim, and Facundo Mémoli;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 34; pp. 34:1–34:17
Leibniz International Proceedings in Informatics

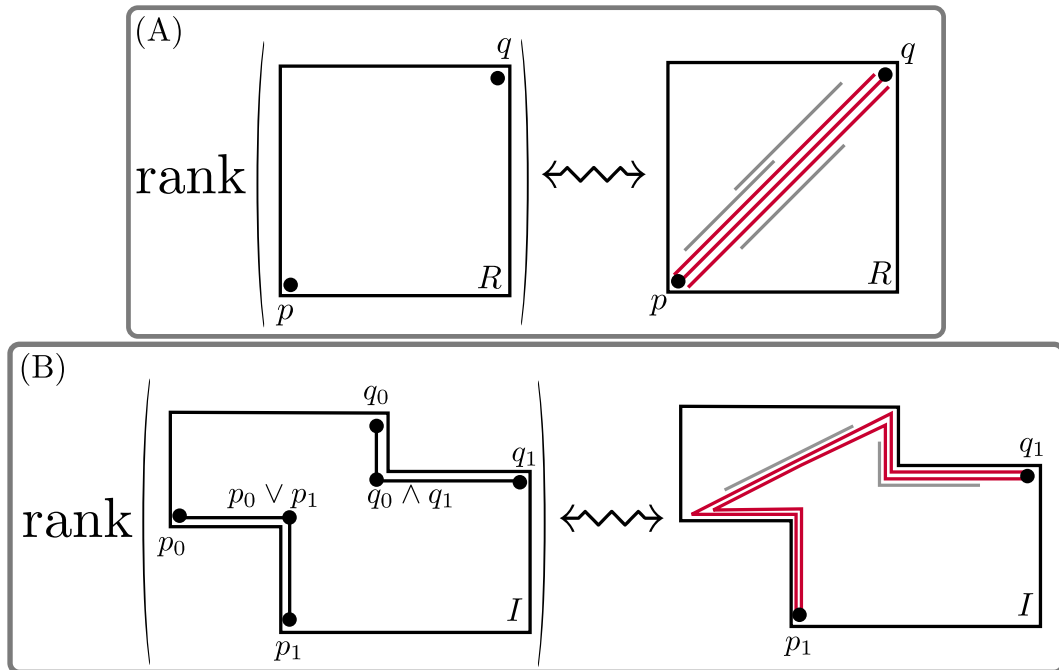


LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



There are many situations in which data naturally induce persistence modules over posets which are different from \mathbf{Z} [4, 12, 13, 14, 17, 22, 27, 29, 30, 33]. Unfortunately, the situation already becomes “wild” when the domain poset is \mathbf{Z}^2 . In that situation, one must contend with the fact that a direct analogue of the notion of persistence diagrams may not exist [14], namely it may not be possible to obtain a lossless up-to-isomorphism representation of the module as a direct sum of interval modules.

Much energy has been put into finding ways in which one can extract incomplete but still stable invariants from persistence modules $M : \mathbf{Z}^d \rightarrow \mathbf{vec}$ (which we will refer to as a \mathbf{Z}^d -module). Biasotti et al. [6] proposed considering the restriction of a \mathbf{Z}^d -module to lines with positive slope. This was further developed by Lesnick and Wright in the RIVET project [31] which facilitates the interactive visualization of \mathbf{Z}^2 -modules. Cai et al. [11] considered a certain elder-rule on the \mathbf{Z}^2 -modules which arise in multiparameter clustering. Other efforts have identified algebraic conditions which can guarantee that M can be decomposed into interval modules of varying degrees of complexity (e.g. rectangle modules etc) [7, 16].



■ **Figure 1 Generalized rank via zigzag persistence.** Let M be a \mathbf{Z}^2 -module. (A) Standard rank: Let $p \leq q$ in \mathbf{Z}^2 . The rank of the structure map from p to q coincides with the multiplicity of full bars (red) over the diagonal path, which is three. (B) Generalized rank: Let I be an interval of \mathbf{Z}^2 . Let us consider the zigzag poset $\partial I : p_1 \leq (p_0 \vee p_1) \geq p_0 \leq q_0 \geq (q_0 \wedge q_1) \leq q_1$. The generalized rank of M over I is equal to the multiplicity of full bars (red) in the zigzag module $M_{\partial I}$, which is two (Theorem 24) (note: by definition, the zigzag poset ∂I does not fully inherit the partial order on \mathbf{Z}^2 . For example, the partial order on ∂I does not contain the pair (p_1, q_1) whereas $p_1 \leq q_1$ in \mathbf{Z}^2).

A distinct thread has been proposed by Patel in [35] through the reinterpretation of the persistence diagram of a \mathbf{Z} -module as the Möbius inversion of its rank function. Patel’s work was then extended by Kim and Mémoli [26] to the setting of modules defined over any suitable locally finite poset. They generalized the rank invariant via the *limit-to-colimit map* over subposets and then conveniently expressed its Möbius inversion. In fact the limit-to-colimit map was suggested by Amit Patel to the authors of [26] who in [25] used it to define a notion

of rank invariant for zigzag modules. Chambers and Letscher [15] also considered a notion of persistent homology over directed acyclic graphs using the limit-to-colimit map. Asashiba et al. [2] study the case of modules defined on an $m \times n$ grid and propose a high-level algorithm for computing both their generalized rank function and their Möbius inversions with the goal of providing an approximation of a given module by interval decomposables. Asashiba et al. [1] tackle the interval decomposability of a given \mathbf{Z}^d -module via quiver representation theory.

One fundamental algorithmic problem is that of determining whether a given \mathbf{Z}^2 -module is interval decomposable, and if so, computing the intervals. There are some existing solutions to this problem in the literature. Suppose that the input \mathbf{Z}^2 -module is induced by a bifiltration comprising at most t simplices on a grid of cardinality $O(t)$. First, the decomposition algorithm by Dey and Xin [20] can produce all indecomposables from such a module in $O(t^{2\omega+1})$ time (see [24] for comments about its implementation) where $\omega \in [2, 2.373]$ is the exponent of matrix multiplication. Given these indecomposables, one could then test whether they are indeed interval modules. However, the algorithm requires that the input module be such that no two generators or relations in the module have the same grade. Then, Asashiba et al. [1] give an algorithm which requires enumerating an exponential number (in t) of intervals. Finally, the algorithm by Meataxe sidesteps both of the above issues, but incurs a worst-case cost of $O(t^{18})$ as explained in [20].

See also [5, 9, 10, 28, 32] for related recent work.

Contributions. One of our key results is the following. We prove that for an interval I in \mathbf{Z}^2 we can compute the generalized rank invariant $\text{rk}(M)(I)$ of a \mathbf{Z}^2 -module M through the computation of the zigzag persistence barcode of the *restriction* of M to the *boundary cap* of I , which is a certain zigzag path in I ; see Figure 1 for an illustration.

These are our main results assuming that the input is a bifiltration with $O(t)$ simplices:

1. We reduce the problem of computing the generalized rank invariant of a \mathbf{Z}^2 -module to computing zigzag persistence (Theorem 24).
2. We provide an algorithm INTERVAL (page 13) to compute the barcode of any finite interval decomposable \mathbf{Z}^2 -module in time $O(t^{\omega+2})$ (Proposition 38).
3. We provide an algorithm ISINTERVALDECOMP (page 15) to decide the interval decomposability of a finite \mathbf{Z}^2 -module in time $O(t^{3\omega+2})$ (Proposition 39).

2 Preliminaries

In §2.1, we review the notion of interval decomposability of persistence modules. In §2.2, we review the notions of generalized rank invariant and generalized persistence diagram. In §2.3, we discuss how to compute the limit and the colimit of a given functor $P \rightarrow \mathbf{vec}$.

2.1 Persistence Modules and their decompositions

We fix a certain field \mathbb{F} and every vector space in this paper is over \mathbb{F} . Let \mathbf{vec} denote the category of *finite dimensional* vector spaces and linear maps over \mathbb{F} .

Let P be a poset. We regard P as the category that has points of P as objects. Also, for any $p, q \in P$, there exists a unique morphism $p \rightarrow q$ if and only if $p \leq q$. For a positive integer d , let \mathbf{Z}^d be given the partial order defined by $(a_1, a_2, \dots, a_d) \leq (b_1, b_2, \dots, b_d)$ if and only if $a_i \leq b_i$ for $i = 1, 2, \dots, d$.

A (P -indexed) **persistence module** is any functor $M : P \rightarrow \mathbf{vec}$ (which we will simply refer to as a P -module). In other words, to each $p \in P$, a vector space M_p is associated, and to each pair $p \leq q$ in P , a linear map $\varphi_M(p, q) : M_p \rightarrow M_q$ is associated. Importantly, whenever $p \leq q \leq r$ in P , it must be that $\varphi_M(p, r) = \varphi_M(q, r) \circ \varphi_M(p, q)$.

We say that a pair of $p, q \in P$ is **comparable** if either $p \leq q$ or $q \leq p$.

► **Definition 1** ([8]). An **interval** I of P is a subset $I \subseteq P$ such that:

- (i) I is nonempty.
- (ii) If $p, q \in I$ and $p \leq r \leq q$, then $r \in I$.
- (iii) I is **connected**, i.e. for any $p, q \in I$, there is a sequence $p = p_0, p_1, \dots, p_\ell = q$ of elements of I with p_i and p_{i+1} comparable for $0 \leq i \leq \ell - 1$.

By $\mathbf{Int}(P)$ we denote the set of all finite intervals of P . When P is finite and connected, $P \in \mathbf{Int}(P)$ will be referred to as **the full interval**.

For an interval I of P , the **interval module** $\mathbb{I}_I : P \rightarrow \mathbf{vec}$ is defined as

$$\mathbb{I}_I(p) = \begin{cases} \mathbb{F} & \text{if } p \in I, \\ 0 & \text{otherwise,} \end{cases} \quad \varphi_{\mathbb{I}_I}(p, q) = \begin{cases} \text{id}_{\mathbb{F}} & \text{if } p, q \in I, p \leq q, \\ 0 & \text{otherwise.} \end{cases}$$

Direct sums and quotients of P -modules are defined pointwisely at each index $p \in P$.

► **Definition 2.** Let M be any P -module. A **submodule** N of M is defined by subspaces $N_p \subseteq M_p$ such that $\varphi_M(p, q)(N_p) \subseteq N_q$ for all $p, q \in P$ with $p \leq q$. These conditions guarantee that N itself is a P -module, with the structure maps given by the restrictions $\varphi_M(p, q)|_{N_p}$. In this case we write $N \leq M$.

A submodule N is a **summand** of M if there exists a submodule N' which is complementary to N , i.e. $M_p = N_p \oplus N'_p$ for all p . In that case, we say that M is a direct sum of N, N' and write $M \cong N \oplus N'$. Note that this direct sum is an internal direct sum.

► **Definition 3.** A P -module M is called **interval decomposable** if M is isomorphic to a direct sum of interval modules, i.e. there exists an indexing set \mathcal{J} such that $M \cong \bigoplus_{j \in \mathcal{J}} \mathbb{I}_{I_j}$ (external direct sum). In this case, the multiset $\{I_j : j \in \mathcal{J}\}$ is called the **barcode** of M , which will be denoted by $\text{barc}(M)$.

The Azumaya-Krull-Remak-Schmidt theorem guarantees that $\text{barc}(M)$ is well-defined [3]. Consider a **zigzag poset** of n points, $\bullet_1 \leftrightarrow \bullet_2 \leftrightarrow \dots \leftrightarrow \bullet_{n-1} \leftrightarrow \bullet_n$ where \leftrightarrow stands for either \leq or \geq . A functor from a zigzag poset to \mathbf{vec} is called a **zigzag module** [12]. Any zigzag module is interval decomposable [23] and thus admits a barcode.

The following proposition directly follows from the Azumaya-Krull-Remak-Schmidt theorem and will be useful in §4.

► **Proposition 4.** Let $M : P \rightarrow \mathbf{vec}$ be interval decomposable and let $N \leq M$ is a summand of M (Definition 2). Then, M/N is interval decomposable (proof in the full version).

2.2 Generalized rank invariant and generalized persistence diagrams

Let P be a finite connected poset and consider any P -module M . Then M admits a limit $\varprojlim M = (L, (\pi_p : L \rightarrow M_p)_{p \in P})$ and a colimit $\varinjlim M = (C, (i_p : M_p \rightarrow C)_{p \in P})$; see the full version for definitions. This implies that, for every $p \leq q$ in P , $\varphi_M(p \leq q) \circ \pi_p = \pi_q$ and $i_q \circ \varphi_M(p \leq q) = i_p$, which in turn imply $i_p \circ \pi_p = i_q \circ \pi_q : L \rightarrow C$ for any $p, q \in P$.

► **Definition 5** ([26]). The **canonical limit-to-colimit map** $\psi_M : \varprojlim M \rightarrow \varinjlim M$ is the linear map $i_p \circ \pi_p$ for any $p \in P$. The **generalized rank** of M is $\text{rank}(M) := \text{rank}(\psi_M)$.

The rank of M counts the multiplicity of the fully supported interval modules in a direct sum decomposition of M .

► **Theorem 6** ([15, Lemma 3.1]). The rank of M is equal to the number of indecomposable summands of M which are isomorphic to the interval module \mathbb{I}_P .

► **Definition 7.** The **(Int)-generalized rank invariant** of M is the map $\text{rk}_{\mathbb{I}}(M) : \mathbf{Int}(P) \rightarrow \mathbf{Z}_+$ defined as $I \mapsto \text{rank}(M|_I)$, where $M|_I$ is the restriction of M to I .

► **Definition 8.** The **(Int)-generalized persistence diagram** of M is the unique¹ function $\text{dgm}_{\mathbb{I}}(M) : \mathbf{Int}(P) \rightarrow \mathbf{Z}$ that satisfies, for any $I \in \mathbf{Int}(P)$,

$$\text{rk}_{\mathbb{I}}(M)(I) = \sum_{\substack{J \supseteq I \\ J \in \mathbf{Int}(P)}} \text{dgm}_{\mathbb{I}}(M)(J).$$

The following is a slight variation of [26, Theorem 3.14] and [2, Theorem 5.10].

► **Theorem 9.** If a given $M : P \rightarrow \mathbf{vec}$ is interval decomposable, then for all $I \in \mathbf{Int}(P)$, $\text{dgm}_{\mathbb{I}}(M)(I)$ is equal to the multiplicity of I in $\text{barc}(M)$ (proof in the full version).

We consider P to be a 2d-grid and focus on the setting of \mathbf{Z}^2 -modules.

► **Definition 10.** For any $I \in \mathbf{Int}(\mathbf{Z}^2)$, we define $\text{nb}_{\mathbb{I}}(I) := \{p \in \mathbf{Z}^2 \setminus I : I \cup \{p\} \in \mathbf{Int}(\mathbf{Z}^2)\}$.

Note that $\text{nb}_{\mathbb{I}}(I)$ is nonempty [2, Proposition 3.2]. When $A \subseteq \text{nb}_{\mathbb{I}}(I)$ contains more than one point, $A \cup I$ is not necessarily an interval of \mathbf{Z}^2 . However, there always exists a unique smallest interval that contains $A \cup I$ which is denoted by $\overline{A \cup I}$.

► **Remark 11** ([2, Theorem 5.3]). If in Definition 8 we assume that $P \in \mathbf{Int}(\mathbf{Z}^2)$ then we have that for every $I \in \mathbf{Int}(P)$,²

$$\text{dgm}_{\mathbb{I}}(M)(I) = \text{rk}_{\mathbb{I}}(M)(I) + \sum_{\substack{A \subseteq \text{nb}_{\mathbb{I}}(I) \cap P \\ A \neq \emptyset}} (-1)^{|A|} \text{rk}_{\mathbb{I}}(M)\left(\overline{A \cup I}\right). \tag{1}$$

2.3 Canonical constructions of limits and colimits

Let M be any P -module.

► **Notation 12.** Let $p, q \in P$ and let $v_p \in M_p$ and $v_q \in M_q$. We write $v_p \sim v_q$ if p and q are comparable, and either v_p is mapped to v_q via $\varphi_M(p, q)$ or v_q is mapped to v_p via $\varphi_M(q, p)$.

The following proposition gives a standard way of constructing a limit and a colimit of a P -module M . Since it is well-known, we do not prove it (see for example [26, Section E]).

► **Proposition 13.**

(i) The limit of M is (isomorphic to) the pair $(W, (\pi_p)_{p \in P})$ where:

$$W := \left\{ (v_p)_{p \in P} \in \bigoplus_{p \in P} M_p : \forall p \leq q \text{ in } P, v_p \sim v_q \right\} \tag{2}$$

and for each $p \in P$, the map $\pi_p : W \rightarrow M_p$ is the canonical projection. An element of W is called a **section** of M .

(ii) The colimit of M is (isomorphic to) the pair $(U, (i_p)_{p \in P})$ described as follows: For $p \in P$, let the map $j_p : M_p \hookrightarrow \bigoplus_{p \in P} M_p$ be the canonical injection. U is the quotient $(\bigoplus_{p \in P} M_p) / T$, where T is the subspace of $\bigoplus_{p \in P} M_p$ which is generated by the vectors of the form $j_p(v_p) - j_q(v_q)$, $v_p \sim v_q$, the map $i_p : M_p \rightarrow U$ is the composition $\rho \circ j_p$, where ρ is the quotient map $\bigoplus_{p \in P} M_p \rightarrow U$.

¹ The existence and uniqueness is guaranteed by properties of the Möbius inversion formula [36, 37].

² In [2], only the case $P = \{1, \dots, m\} \times \{1, \dots, n\} \subset \mathbf{Z}^2$ was considered. However, it is not difficult to check that Eq. (1) is still valid for any finite interval P in \mathbf{Z}^2 and any subinterval $I \subseteq P$.

► **Setup 1.** In the rest of the paper, limits and colimits of a P -module M will all be constructed as in Proposition 13. Hence, assuming that P is connected, the canonical limit-to-colimit map $\varprojlim M \rightarrow \varinjlim M$ is $\psi_M := i_p \circ \pi_p$ for any $p \in P$.

3 Computing generalized rank via boundary zigzags

In §3.1 we introduce the notions of lower and upper fences of a poset. In §3.2, we introduce the *boundary cap* ∂I of a finite interval I of \mathbf{Z}^2 , which is a path, a certain sequence of points in I . In §3.3, we show that the rank of any functor $M : I \rightarrow \mathbf{vec}$ can be obtained by computing the barcode of the zigzag module over the path ∂I .

3.1 Lower and upper fences of a poset

Let P be any connected poset. Given any $p \in P$, by p^\downarrow , we denote the set of all elements of P that are less than or equal to p . Dually p^\uparrow is defined as the set of all elements of P that are greater than or equal to p .

► **Definition 14.** A subposet $L \subset P$ (resp. $U \subset P$) is called a lower (resp. upper) fence of P if L is connected, and for any $q \in P$, the intersection $L \cap q^\downarrow$ (resp. $U \cap q^\uparrow$) is nonempty and connected.

► **Proposition 15.** Let L and U be a lower and an upper fences of P respectively. Given any P -module M , we have $\varprojlim M \cong \varprojlim M|_L$ and $\varinjlim M \cong \varinjlim M|_U$ (proof in the full version).³

The canonical isomorphism $\varprojlim M \cong \varprojlim M|_L$ in Proposition 15 is given by the canonical section extension $e : \varprojlim M|_L \rightarrow \varprojlim M$. Namely,

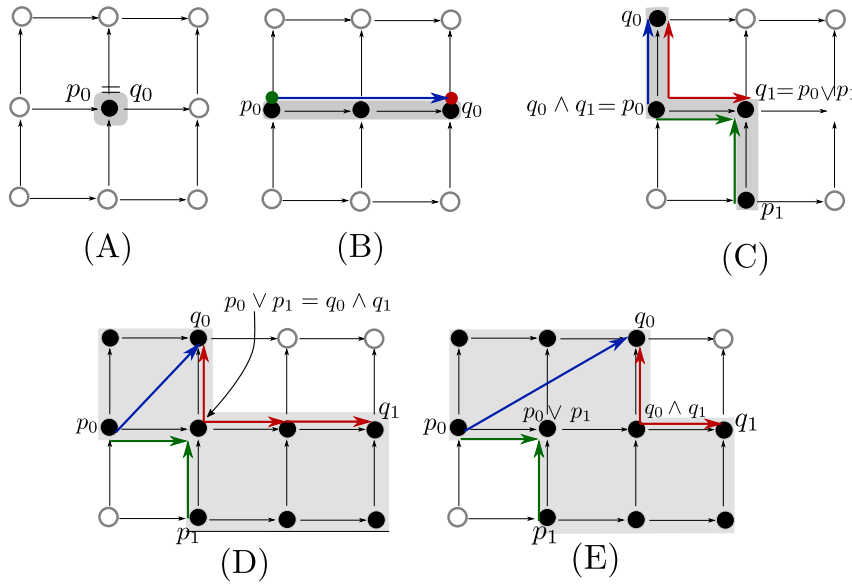
$$e : (\mathbf{v}_p)_{p \in L} \mapsto (\mathbf{w}_q)_{q \in P}, \quad (3)$$

where for any $q \in P$, the vector \mathbf{w}_q is defined as $\varphi_M(p, q)(\mathbf{v}_p)$ for any $p \in L \cap q^\downarrow$; the connectedness of $L \cap q^\downarrow$ guarantees that \mathbf{w}_q is well-defined. Also, if $q \in L$, then $\mathbf{w}_q = \mathbf{v}_q$. The inverse $r := e^{-1}$ is the canonical section restriction. The other isomorphism $\varinjlim M \cong \varinjlim M|_U$ in Proposition 15 is given by the map $i : \varinjlim M|_U \rightarrow \varinjlim M$ defined by $[v_p] \mapsto [v_p]$ for any $p \in U$ and any $v_p \in M_p$; the fact that this map i is well-defined will become clear from Proposition 22. Let us define $\xi : \varprojlim M|_L \rightarrow \varinjlim M|_U$ by $i^{-1} \circ \psi_M \circ e$. By construction, the following diagram commutes

$$\begin{array}{ccc} \varprojlim M|_L & \xrightarrow{\xi} & \varinjlim M|_U \\ \cong \downarrow e & & \cong \downarrow i \\ \varprojlim M & \xrightarrow{\psi_M} & \varinjlim M, \end{array} \quad (4)$$

where ψ_M is the canonical limit-to-colimit map of M . Hence we have the fact $\text{rank}(\psi_M) = \text{rank}(\xi)$, which is useful for proving Theorem 24.

³ This proposition appeared in an earlier version of [26] (see Proposition D.14 in the second arXiv version).



■ **Figure 2** Five different intervals I of \mathbf{Z}^2 . Relations in $\min_{\mathbf{ZZ}}(I)$ and $\max_{\mathbf{ZZ}}(I)$ are indicated by green and red arrows, respectively. The inequality $p_0 \leq q_0$ is indicated by blue arrows unless $p_0 = q_0$. Notice that ∂I , as defined in equation (7), has cardinality 2, 2, 6, 6, 6 in that order ((A),(B),(C),(D),(E)).

3.2 Boundary cap of an interval in \mathbf{Z}^2

Let $I \in \text{Int}(\mathbf{Z}^2)$, i.e. I is a finite interval of \mathbf{Z}^2 (Definition 1). By $\min(I)$ and $\max(I)$, we denote the collections of minimal and maximal elements of I , respectively. In other words,

$$\min(I) := \{p \in I : \text{there is no } q \in I \text{ s.t. } q < p\},$$

$$\max(I) := \{p \in I : \text{there is no } q \in I \text{ s.t. } p < q\}.$$

Note that $\min(I)$ and $\max(I)$ are nonempty and that $\min(I)$ and $\max(I)$ respectively form an *antichain* in I , i.e. any two different points in $\min(I)$ (or in $\max(I)$) are *not* comparable.

► Remark 16.

- (i) The least upper bound and the greatest lower bound of $p, q \in \mathbf{Z}^2$ are denoted by $p \vee q$ and $p \wedge q$ respectively. Let $p = (p_x, p_y)$ and $q = (q_x, q_y)$ in \mathbf{Z}^2 . Then,

$$p \vee q = (\max\{p_x, q_x\}, \max\{p_y, q_y\}), \quad p \wedge q = (\min\{p_x, q_x\}, \min\{p_y, q_y\}).$$

For the item below, let $I \in \text{Int}(\mathbf{Z}^2)$. Notice the following:

- (ii) Since $\min(I)$ is a finite antichain, we can list the elements of $\min(I)$ in ascending order of their x -coordinates, i.e. $\min(I) := \{p_0, \dots, p_k\}$ and such that for each $i = 0, \dots, k$, the x -coordinate of p_i is less than that of p_{i+1} . Similarly, let $\max(I) := \{q_0, \dots, q_\ell\}$ be ordered in ascending order of q_j 's x -coordinates. We have that $p_0 \leq q_0$ (Figure 2).

► **Definition 17** (Lower and upper zigzags of an interval). *Let I , $\min(I)$, and $\max(I)$ be as in Remark 16 ii. We define the following two zigzag posets (Figure 2):*

$$\begin{aligned} \min_{\mathbf{ZZ}}(I) &:= \{p_0 < (p_0 \vee p_1) > p_1 < (p_1 \vee p_2) > \dots < (p_{k-1} \vee p_k) > p_k\} \\ &= \min(I) \cup \{p_i \vee p_{i+1} : i = 0, \dots, k-1\}, \end{aligned} \tag{5}$$

$$\begin{aligned} \max_{\mathbf{ZZ}}(I) &:= \{q_0 > (q_0 \wedge q_1) < q_1 > (q_1 \wedge q_2) < \dots > (q_{\ell-1} \wedge q_\ell) > q_\ell\} \\ &= \max(I) \cup \{q_i \wedge q_{i+1} : i = 0, \dots, \ell-1\}. \end{aligned} \tag{6}$$

Note that $\min_{\mathbf{ZZ}}(I)$ and $\max_{\mathbf{ZZ}}(I)$ are lower and upper fences of I respectively.

For $p, q \in P$, let us write $p \triangleleft q$ if $p < q$ and there is no $r \in P$ such that $p < r < q$. Similarly, we write $p \triangleright q$ if $p > q$ and there is no $r \in P$ such that $p > r > q$.

► **Definition 18.** Given a poset P , a **path** Γ between two points $p, q \in P$ is a sequence of points $p = p_0, \dots, p_k = q$ in P such that either $p_i \leq p_{i+1}$ or $p_i \geq p_{i+1}$ for every $i \in [1, k-1]$ (in particular, there can be a pair $i \neq j$ such that $p_i = p_j$). The path Γ is said to be **monotonic** if $p_i \leq p_{i+1}$ for each i . The path Γ is called **faithful** if either $p_i \triangleleft p_{i+1}$ or $p_i \triangleright p_{i+1}$ for each i .

► **Definition 19** (Boundary cap of an interval). We define the **boundary cap** ∂I of $I \in \text{Int}(\mathbf{Z}^2)$ as the path obtained by concatenating $\min_{\mathbf{ZZ}}(I)$ and $\max_{\mathbf{ZZ}}(I)$ in Eqs. (5) and (6).

$$\partial I := \underbrace{p_k < (p_k \vee p_{k-1}) > p_{k-1} < \cdots > p_0}_{2k+1 \text{ terms from } \min_{\mathbf{ZZ}}(I)} \leq \underbrace{q_0 > (q_0 \wedge q_1) < q_1 > \cdots < q_\ell}_{2\ell+1 \text{ terms from } \max_{\mathbf{ZZ}}(I)}, \quad (7)$$

We remark that ∂I can contain multiple copies of the same point. Namely, there can be $i \in [0, k]$ and $j \in [0, \ell]$ such that either $p_i = q_j$ (Figure 2 (A)), $p_i = q_j \wedge q_{j+1}$ (Figure 2 (C)), $p_i \vee p_{i+1} = q_j$ (Figure 2 (C)), or $p_i \vee p_{i+1} = q_j \wedge q_{j+1}$ (Figure 2 (D)).

Consider the following zigzag poset of the same length as ∂I :

$$\mathbf{ZZ}_{\partial I} : \underbrace{\bullet_1 < \bullet_2 > \bullet_3 < \cdots > \bullet_{2k+1}}_{2k+1} < \underbrace{\circ_1 > \circ_2 < \circ_3 > \cdots < \circ_{2\ell+1}}_{2\ell+1}. \quad (8)$$

Still using the notation in Eqs. (7) we have the following order-preserving map

$$\iota_I : \mathbf{ZZ}_{\partial I} \rightarrow I \quad (9)$$

whose image is ∂I : \bullet_1 is sent to p_k , \bullet_2 is sent to $p_k \vee p_{k-1}$, \dots , and $\circ_{2\ell+1}$ is sent to q_ℓ .

3.3 Generalized rank invariant via boundary zigzags

The goal of this section is to establish Theorem 24.

► **Definition 20.** Let P be a poset. Let $\Gamma : p_0, \dots, p_k$ be a path in P . A $(k+1)$ -tuple $\mathbf{v} \in \bigoplus_{i=0}^k M_{p_i}$ is called the **section of M along Γ** if $\mathbf{v}_{p_i} \sim \mathbf{v}_{p_{i+1}}$ for each i (Notation 12).

Note that \mathbf{v} is not necessarily a section of the restriction $M|_{\{p_0, \dots, p_k\}}$ of M to the subposet $\{p_0, \dots, p_k\} \subseteq I$. Furthermore, Γ can contain multiple copies of the same point in P .

► **Example 21.** Consider $M : \{(1, 1), (1, 2), (2, 2), (2, 1)\} \subset \mathbf{Z}^2 \rightarrow \mathbf{vec}$ given as follows.

$$\begin{array}{ccc} M_{(1,2)} & \longrightarrow & M_{(2,2)} \\ \uparrow & & \uparrow \\ M_{(1,1)} & \longrightarrow & M_{(2,1)} \end{array} = \begin{array}{ccc} \mathbb{F} & \xrightarrow{1} & \mathbb{F} \\ \uparrow 1 & & \uparrow (1 \ 1) \\ \mathbb{F} & \xrightarrow{\begin{pmatrix} 1 \\ 0 \end{pmatrix}} & \mathbb{F}^2 \end{array}$$

Consider the path $\Gamma : (1, 1), (1, 2), (2, 2), (2, 1)$ which contains all points in the indexing poset. Then, $\mathbf{v} := (1, 1, 1, (0, 1)) \in M_{(1,1)} \oplus M_{(1,2)} \oplus M_{(2,2)} \oplus M_{(2,1)}$ is a section of M along Γ , while \mathbf{v} is *not* a section of M itself, i.e. $\mathbf{v} \notin \varprojlim M$.

By Proposition 13 (ii), we directly have:

► **Proposition 22.** *Let $p, q \in P$. For any vectors $v_p \in M_p$ and $v_q \in M_q$ $[v_p] = [v_q]$ in⁴ the colimit $\varinjlim M$ if and only if there exist a path $\Gamma : p = p_0, p_1, \dots, p_n = q$ in P and a section \mathbf{v} of M along Γ such that $\mathbf{v}_p = v_p$ and $\mathbf{v}_q = v_q$.*

The map $\iota_I : \mathbf{ZZ}_{\partial I} \rightarrow I$ in Eq.s. (9) induces a bijection between the sections of $M_{\partial I}$ and the sections of M along ∂I in a canonical way. Hence:

► **Setup 2.** In the rest of §3.3, we fix both $I \in \mathbf{Int}(\mathbf{Z}^2)$ and a functor $M : I \rightarrow \mathbf{vec}$. Each element in $\varinjlim M_{\partial I}$ is identified with the corresponding section of M along ∂I . Also, we identify points in (7) and (8) via ι_I .

► **Definition 23** (Zigzag module along ∂I). *Define the zigzag module $M_{\partial I} : \mathbf{ZZ}_{\partial I} \rightarrow \mathbf{vec}$ by $(M_{\partial I})_x := M_{\iota_I(x)}$ for $x \in \mathbf{ZZ}_{\partial I}$ and $\varphi_{M_{\partial I}}(x, y) := \varphi_M(\iota_I(x), \iota_I(y))$ for $x \leq y$ in $\mathbf{ZZ}_{\partial I}$.*

One of our main results is the following.

► **Theorem 24.** *$\text{rank}(M)$ is equal to the multiplicity of the full interval in $\text{barc}(M_{\partial I})$.*

Proof. By Theorem 6, it suffices to show that

$$\text{rank}(\psi_M : \varinjlim M \rightarrow \varinjlim M) = \text{rank}(\psi_{M_{\partial I}} : \varinjlim M_{\partial I} \rightarrow \varinjlim M_{\partial I}).$$

Let $L := \min_{\mathbf{ZZ}}(I)$ and $U := \max_{\mathbf{ZZ}}(I)$ which are lower and upper fences of I respectively. Let us define the maps e, r, i and ξ as described in the paragraph after Proposition 15. Then, by Proposition 15 and the commutative diagram in (4), it suffices to prove that the rank of ξ equals the rank of $\psi_{M_{\partial I}}$. To this end, we show that there exist a surjective linear map $f : \varinjlim M_{\partial I} \rightarrow \varinjlim M|_L$ and an injective linear map $g : \varinjlim M|_U \rightarrow \varinjlim M_{\partial I}$ such that $\psi_{M_{\partial I}} = g \circ \xi \circ f$. We define f as the canonical section restriction $(\mathbf{v}_q)_{q \in \partial I} \mapsto (\mathbf{v}_q)_{q \in L}$. We define g as the canonical map, i.e. $[v_q] \mapsto [v_q]$ for any $q \in U$ and any $v_q \in M_q$. By Proposition 22 and by construction of $M_{\partial I}$, the map g is well-defined.

We now show that $\psi_{M_{\partial I}} = g \circ \xi \circ f$. Let $\mathbf{v} := (\mathbf{v}_q)_{q \in \partial I} \in \varinjlim M_{\partial I}$. Then, by definition of $\psi_{M_{\partial I}}$ (Setup 1), the image of \mathbf{v} via $\psi_{M_{\partial I}}$ is $[v_{q_0}]$ where $q_0 \in U$ is defined as in Remark 16 ii. Also, we have

$$\mathbf{v} \xrightarrow{f} (\mathbf{v}_q)_{q \in L} \xrightarrow{\xi} [v_{q_0}] (\in \varinjlim M|_U) \xrightarrow{g} [v_{q_0}] (\in \varinjlim M_{\partial I}),$$

which proves the equality $\psi_{M_{\partial I}} = g \circ \xi \circ f$.

We claim that f is surjective. Let $r' : \varinjlim M \rightarrow \varinjlim M_{\partial I}$ be the canonical section restriction map $(\mathbf{v}_q)_{q \in I} \mapsto (\mathbf{v}_q)_{q \in \partial I}$. Then, the restriction $r : \varinjlim M \rightarrow \varinjlim M|_L$, can be seen as the composition of two restrictions $r = f \circ r'$. Since r is the inverse of the isomorphism e in diagram (4), r is surjective and thus so is f .

Next we claim that g is injective. Let $i' : \varinjlim M_{\partial I} \rightarrow \varinjlim M$ be defined by $[v_q] \mapsto [v_q]$ for any $q \in \partial I$ and any $v_q \in M_q$. By Proposition 22 and by construction of $M_{\partial I}$, the map i' is well-defined. Then, for the isomorphism i in diagram (4), we have $i = i' \circ g$. This implies that g is injective. ◀

⁴ For simplicity, we write $[v_p]$ and $[v_q]$ instead of $[j_p(v_p)]$ and $[j_q(v_q)]$ respectively where $j_p : M_p \rightarrow \bigoplus_{r \in P} M_r$ and $j_q : M_q \rightarrow \bigoplus_{r \in P} M_r$ are the canonical inclusion maps.

34:10 Generalized Rank via Zigzag and Its Applications

► **Remark 25.** In Definition 19 one may consider the “lower” boundary cap $\widehat{\partial I}$, as an alternative to ∂I :

$$\widehat{\partial I} : p_0 < p_0 \vee p_1 > p_1 < \cdots > p_k \leq q_\ell > q_\ell \wedge q_{\ell-1} < q_{\ell-1} > \cdots < q_0.$$

The value $\text{rank}(M)$ also equals the multiplicity of the full interval in the barcode of the zigzag module induced over $\widehat{\partial I}$.

By Theorem 24, we can utilize algorithms for zigzag persistence in order to compute the generalized rank invariant and the generalized persistence diagram of any \mathbf{Z}^2 -module that is obtained by applying the homology functor to a finite simplicial bifiltration consisting of $O(t)$ simplices over an index set of size $O(t)$. For this, we complete the boundary cap of a given interval to a faithful path (i.e. we put the missing monotonic paths between every pair of consecutive points) and then simply run a zigzag persistence algorithm, say the $O(t^\omega)$ algorithm of Milosavljevic et al. [34], on the filtration restricted to this path.

► **Remark 26.** To compute $\text{dgm}_{\mathbb{I}}(M)(I)$ by the formula in (1), one needs to consider terms whose number depends exponentially on the number of neighbors of I . However, for any interval that has at most $O(\log t)$ neighbors, we have $2^{O(\log t)} = t^c$ terms for some constant $c > 0$. It follows that using $O(t^\omega)$ zigzag persistence algorithm for computing generalized ranks, we obtain an $O(t^{\omega+c})$ algorithm for computing generalized persistence diagrams of intervals that have at most $O(\log t)$ neighbors.

4 Computing intervals and detecting interval decomposability

When a persistence module M admits a summand N that is isomorphic to an interval module, N will be called an **interval summand** of M . In this section, we apply Theorem 24 for computing generalized rank via zigzag to different problems that ask to find interval summands of an input finite \mathbf{Z}^2 -module: Problems I, II, and III.

Let K be a finite abstract simplicial complex and let $\text{sub}(K)$ be the poset of all subcomplexes of K , ordered by inclusion. Given any poset P , an order-preserving map $\mathcal{F} : P \rightarrow \text{sub}(K)$ is called a simplicial filtration (of K).

► **Setup 3.** Throughout §4, \mathcal{F} denotes a bifiltration of a simplicial complex K defined over an interval $P \in \mathbf{Int}(\mathbf{Z}^2)$. Let $t := \max(|K|, |P|)$ denote the maximum of the number of simplices in K and the number of points in P . By $M_{\mathcal{F}} : P \rightarrow \mathbf{vec}$ we denote the module induced by \mathcal{F} through the homology functor with coefficients in the field \mathbb{F} .

Computing the dimension function. In all algorithms below, we utilize a subroutine $\text{DIM}(\mathcal{F}, P)$, which computes the dimension of the vector space $(M_{\mathcal{F}})_p$ for every $p \in P$.

► **Proposition 27.** $\text{DIM}(\mathcal{F}, P)$ can be executed in $O(t^3)$ time (proof in the full version).

4.1 Detecting interval modules

We consider the following problem.

► **Problem I.** Determine whether $M_{\mathcal{F}}$ is isomorphic to the direct sum of a certain number of copies of \mathbb{I}_P and if so, report the number of such copies.

Algorithm ISINTERVAL solves Problem I. The correctness of the algorithm follows from Proposition 28. Below, for an interval $I \in \mathbf{Int}(\mathbf{Z}^2)$ and for $m \in \mathbf{Z}_{\geq 0}$ we define $\mathbb{I}_I^m := \underbrace{\mathbb{I}_I \oplus \mathbb{I}_I \oplus \cdots \oplus \mathbb{I}_I}_m$. In particular, \mathbb{I}_I^0 is defined to be the trivial module. Let us recall that $(M_{\mathcal{F}})_{\partial P}$ denotes the zigzag module along the boundary cap ∂P (Definition 23).

■ **Algorithm 1** ISINTERVAL(\mathcal{F}, P).

-
- Step 1. Compute zigzag barcode $\text{barc}((M_{\mathcal{F}})_{\partial P})$ and let m be the multiplicity of the full interval.
 - Step 2. Call DIM(\mathcal{F}, P) (Computes $\dim(M_{\mathcal{F}})_p$ for every $p \in P$)
 - Step 3. If $\dim(M_{\mathcal{F}})_p == m$ for each point $p \in P$ return m , otherwise return 0 indicating $M_{\mathcal{F}}$ has a summand which is not an interval module supported over P .
-

► **Proposition 28.** Assume that a given $M : P \rightarrow \mathbf{vec}$ has the indecomposable decomposition $M \cong \bigoplus_{i=1}^m M_i$. Then, every summand M_i is isomorphic to the interval module \mathbb{I}_P if and only if $\text{rk}_{\mathbb{I}}(M)(P) = \dim M_p = m$ for all $p \in P$ (proof in the full version).

► **Proposition 29.** Algorithm ISINTERVAL can be run in $O(t^3)$ time (proof in the full version).

4.2 Interval decomposable modules and its summands

Setup 3 still applies in §4.2. Next, we consider the problem of computing all indecomposable summands of $M_{\mathcal{F}}$ under the assumption that $M_{\mathcal{F}}$ is interval decomposable (Definition 3).

► **Problem II.** Assume that $M_{\mathcal{F}} : P \rightarrow \mathbf{vec}$ is interval decomposable. Find $\text{barc}(M_{\mathcal{F}})$.

We present algorithm INTERVAL to solve Problem II in $O(t^{\omega+2})$ time. This algorithm is eventually used to detect whether a given module is interval decomposable or not (Problem III). Before describing INTERVAL, we first describe another algorithm TRUEINTERVAL. The outcomes of both INTERVAL and TRUEINTERVAL are the same as the barcode of $M_{\mathcal{F}}$ in Problem II (Propositions 33 and 37). Whereas TRUEINTERVAL is more intuitive, real implementation is accomplished via INTERVAL.

► **Definition 30.** Let $\mathcal{I}(M_{\mathcal{F}}) := \{I \in \mathbf{Int}(P) : \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I) > 0\}$. We call $I \in \mathcal{I}(M_{\mathcal{F}})$ maximal if there is no $J \supsetneq I$ in $\mathbf{Int}(P)$ such that $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(J)$ is nonzero.

► **Proposition 31.** Assume that $M_{\mathcal{F}}$ is interval decomposable and let $I \in \mathcal{I}(M_{\mathcal{F}})$ be maximal. Then, I belongs to $\text{barc}(M_{\mathcal{F}})$ and the multiplicity of I in $\text{barc}(M_{\mathcal{F}})$ is equal to $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I)$.

Proof. By assumption, all summands in the sum $\sum_{\substack{A \subseteq \text{nb}_{\mathbb{I}}(I) \cap P \\ A \neq \emptyset}} (-1)^{|A|} \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(\overline{I \cup A})$ corresponding to the second term of (1) are zero. Hence, $\text{dgm}_{\mathbb{I}}(M_{\mathcal{F}})(I) = \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I) > 0$. Since $M_{\mathcal{F}}$ is interval decomposable, by Theorem 9, $\text{dgm}_{\mathbb{I}}(M_{\mathcal{F}})(I)$ is equal to the multiplicity of I in $\text{barc}(M_{\mathcal{F}})$. Therefore, not only does I belong to $\text{barc}(M_{\mathcal{F}})$, but also the value $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I)$ is equal to the multiplicity of I in $\text{barc}(M_{\mathcal{F}})$. ◀

The following proposition is a corollary of Proposition 31.

34:12 Generalized Rank via Zigzag and Its Applications

► **Proposition 32.** *Assume that $M_{\mathcal{F}}$ is interval decomposable and let $I \in \mathcal{I}(M_{\mathcal{F}})$ be maximal. Let $\mu_I := \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I)$. Then, $M_{\mathcal{F}}$ admits a summand N which is isomorphic to $\mathbb{I}_I^{\mu_I}$.*

Let us now describe a procedure `TRUEINTERVAL` that outputs all indecomposable summands of a given interval decomposable module. For computational efficiency, we will implement `TRUEINTERVAL` differently. Let $M := M_{\mathcal{F}}$. First we compute $\dim M_p$ for every point $p \in P$. Iteratively, we choose a point p with $\dim M_p \neq 0$ and compute a maximal interval $I \in \mathcal{I}(M)$ containing p . Since M is interval decomposable, by Propositions 31 and 32 we have that $I \in \text{barc}(M)$ and that there is a summand $N \cong \mathbb{I}_I^{\mu_I}$ of M . Consider the quotient module $M' := M/N$. Clearly, this “peeling off” of N reduces the total dimension of the input module. Namely, $\dim M'_p = \begin{cases} \dim M_p - \mu_I, & p \in I \\ \dim M_p, & p \notin I. \end{cases}$ We continue the process by replacing M with M' until there is no point $p \in P$ with $\dim M_p \neq 0$ (note that M' is interval decomposable by Proposition 4). Since $\dim M := \sum_{p \in P} \dim M_p$ is finite, this process terminates in finitely many steps. By Propositions 4 and 32, the outcome of `TRUEINTERVAL` is a list of all intervals in $\text{barc}(M)$ with accurate multiplicities:

► **Proposition 33.** *Assume that $M_{\mathcal{F}}$ is interval decomposable. Let $I_i, i = 1, \dots, k$ be the intervals computed by `TRUEINTERVAL`. For each $i = 1, \dots, k$, let $\mu_{I_i} := \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I_i)$. Then, we have $M_{\mathcal{F}} \cong \bigoplus_{i=1}^k \mathbb{I}_{I_i}^{\mu_{I_i}}$.*

Next, we describe an algorithm `INTERVAL` that simulates `TRUEINTERVAL` while avoiding explicit quotienting of $M_{\mathcal{F}}$ by its summands.

We associate a number $d(p)$ and a list $\text{list}(p)$ of identifiers of intervals $I \subseteq P$ to each point $p \in P$. The number $d(p)$ equals the original dimension of $(M_{\mathcal{F}})_p$ minus the number of intervals peeled off so far (counted with their multiplicities) which contained p . It is initialized to $\dim(M_{\mathcal{F}})_p$. Each time we compute a maximal interval $I \in \mathcal{I}(M_{\mathcal{F}})$ with multiplicity μ_I that contains p , we update $d(p) := d(p) - \mu_I$ keeping track of how many more intervals containing p would `TRUEINTERVAL` still be peeling off.

With each interval I that is output, we associate an identifier $\text{id}(I)$. The variable $\text{list}(p)$ maintains the set of identifiers of the intervals containing p that have been output so far. While searching for a maximal interval I , we maintain a variable list for I that contains the set of identifiers common to all points in I . Initializing list with $\text{list}(p)$ of the initial point p , we update it as we explore expanding I . Every time we augment I with a new point q , we update list by taking its intersection with the set of identifiers $\text{list}(q)$ associated with q .

We assume a routine `COUNT` that takes a list as input and gives the total number of intervals counted with their multiplicities whose identifiers are in the list. This means that if $\text{list} = \{\text{id}(I_1), \dots, \text{id}(I_k)\}$, then `COUNT(list)` returns the number $c := \sum \mu_{I_1} + \dots + \mu_{I_k}$.

Notice that, while searching for a maximal interval starting from a point, we keep considering the original given module $M_{\mathcal{F}}$ since we do not implement the true “peeling” (i.e. quotient $M_{\mathcal{F}}$ by a submodule). However, we modify the condition for checking the maximality of an interval I . We check whether $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I) > c$, that is, whether the generalized rank of $M_{\mathcal{F}}$ over I is larger than the total number of intervals containing I that would have been peeled off so far by `TRUEINTERVAL`. This idea is implemented in the following algorithm.

■ **Algorithm 2** INTERVAL (\mathcal{F}, P).

-
- Step 1. Call $\text{DIM}(\mathcal{F}, P)$ and set $d(p) := \dim(M_{\mathcal{F}})_p$; $\text{list}(p) := \emptyset$ for every $p \in P$
 - Step 2. While there exists a $p \in P$ with $d(p) > 0$ do
 - Step 2.1 Let $I := \{p\}$; $\text{list} := \text{list}(p)$; unmark every $q \in P$
 - Step 2.2 If there exists unmarked $q \in \text{nb}_{\mathbb{I}}(I)$ then
 - i. $\text{templist} := \text{list} \cap \text{list}(q)$; $c := \text{COUNT}(\text{templist})$
 - ii. If $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I \cup \{q\}) > c$ then⁵ mark q ; set $I := I \cup \{q\}$; $\text{list} := \text{list} \cap \text{list}(q)$
 - iii. go to Step 2.2
 - Step 2.3 Output I with multiplicity $\mu_I := \text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I) - c$
 - Step 2.4 For every $q \in I$ set $d(q) := d(q) - \mu_I$ and $\text{list}(q) := \text{list}(q) \cup \{\text{id}(I)\}$
-

The output of INTERVAL can be succinctly described as:

Output: $\{(I_i, \mu_{I_i}) : i = 1, \dots, k\}$ where $I_i \in \mathbf{Int}(P)$ and μ_{I_i} is a positive integer for each i .

► **Remark 34.** For each $p \in P$, $\dim M_p$ coincides with $\sum_{I_i \ni p} \mu_{I_i}$.

We will show that if $M_{\mathcal{F}}$ is interval decomposable, then the output of INTERVAL coincides with the barcode of $M_{\mathcal{F}}$ (Propositions 33 and 37).

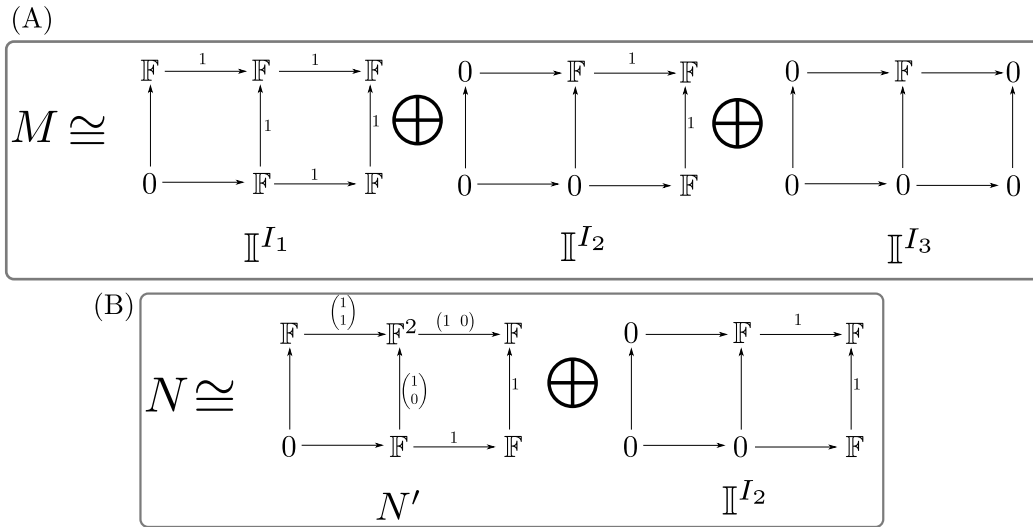
► **Example 35** (INTERVAL with interval decomposable input). Suppose that $M_{\mathcal{F}} \cong \mathbb{I}_{I_1} \oplus \mathbb{I}_{I_2} \oplus \mathbb{I}_{I_3}$ as depicted in Figure 3 (A). The algorithm INTERVAL yields $\{(I_1, 1), (I_2, 1), (I_3, 1)\}$. In particular, since $I_1 \supset I_2 \supset I_3$, INTERVAL outputs $(I_1, 1)$, $(I_2, 1)$, and $(I_3, 1)$ in order, as depicted in Figure 4 (A) (details in the full version).

► **Example 36** (INTERVAL with non-interval-decomposable input). Suppose that $N := M_{\mathcal{F}} \cong N' \oplus \mathbb{I}_{I_2}$ as depicted in Figure 3 (B). N' is an indecomposable module that is not an interval module. One possible final output of INTERVAL is $\{(J_1, 1), (J_2, 1), (J_3, 1)\}$ as depicted in Figure 4 (B). Note however that, depending on the choices of p in Step 2 and the neighbors q in Step 2.2, the final outcome can be different (details in the full version).

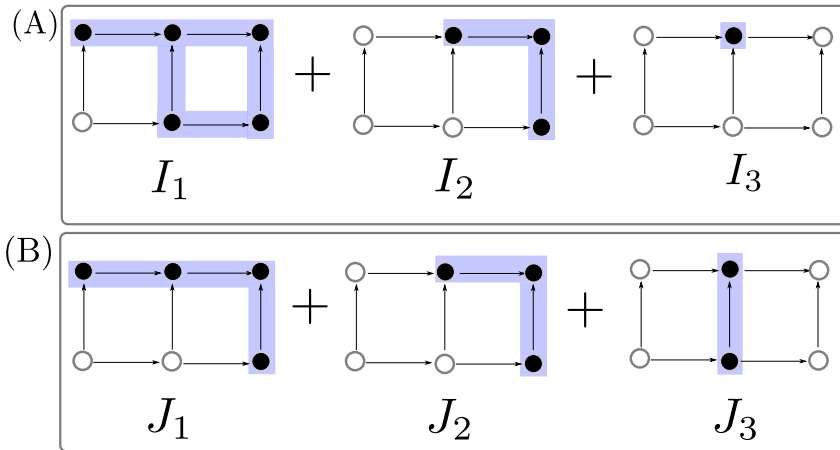
► **Proposition 37.** *If $M_{\mathcal{F}}$ is interval decomposable, INTERVAL(\mathcal{F}, P) computes an interval in $\text{barc}(M_{\mathcal{F}})$ if and only if TRUEINTERVAL(\mathcal{F}, P) computes it with the same multiplicity.*

Proof. (“if”): We induct on the list of intervals in the order they are computed by TRUEINTERVAL. We prove two claims by induction: (i) TRUEINTERVAL can be run to explore the points in P in the same order as INTERVAL while searching for maximal intervals, (ii) if $I_i, i = 1, \dots, k$, are the intervals computed by TRUEINTERVAL with this chosen order, then INTERVAL also outputs these intervals with the same multiplicities. Clearly, for $i = 1$, INTERVAL computes the maximal interval on the same input module $M_{\mathcal{F}}$ as TRUEINTERVAL does. So, clearly, TRUEINTERVAL can be made to explore P as INTERVAL does and hence their outputs are the same. Assume inductively that the hypotheses hold for $i \geq 1$. Then, TRUEINTERVAL operates next on the module $M_{i+1} := M_{\mathcal{F}} / (\mathbb{I}_{I_1}^{\mu_{I_1}} \oplus \dots \oplus \mathbb{I}_{I_i}^{\mu_{I_i}})$ (here each $\mathbb{I}_{I_i}^{\mu_{I_i}}$ stands for a summand of $M_{\mathcal{F}}$ that is isomorphic to $\mathbb{I}_{I_i}^{\mu_{I_i}}$ by Proposition 32). We let TRUEINTERVAL explore P in the same way as INTERVAL does. This is always possible because the outcome of the test for exploration remains the same in both cases as we argue.

⁵ to check $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I \cup \{q\}) > c$, we invoke Theorem 24 and run the zigzag persistence algorithm described beneath Remark 25. For efficiency, one can use zigzag update algorithm in [17].



■ **Figure 3** Modules $M, N : \{1, 2, 3\} \times \{1, 2\} \rightarrow \text{vec}$. M is interval decomposable, but N is not.



■ **Figure 4** An illustration for Examples 35 and 36.

The variable $d(p)$ at this point has the value $\dim(M_{i+1})_p$ and thus both `TRUEINTERVAL` and `INTERVAL` can start exploring from the point p if $d(p) > 0$. So, we let `TRUEINTERVAL` compute the next maximal interval I_{i+1} starting from the point p if `INTERVAL` starts from p .

Now, when `INTERVAL` tests for a point q to expand the interval I , we claim that the result would be the same if `TRUEINTERVAL` tested for q . First of all, the condition whether $I \cup \{q\}$ is an interval or not does not depend on which algorithm we are executing. Second, the list supplied to `COUNT` in Step 2.2 (i) exactly equals the list of intervals containing $I \cup \{q\}$ that `INTERVAL` has already output. By the inductive hypothesis, this list is exactly equal to the list of intervals that `TRUEINTERVAL` had already “peeled off”. Therefore, the test $\text{rk}_{\mathbb{I}}(M_{\mathcal{F}})(I \cup \{q\}) > c$ that `INTERVAL` performs in Step 2.2 (ii) is exactly the same as the test $\text{rk}_{\mathbb{I}}(M_{i+1})(I \cup \{q\}) > 0$ that `TRUEINTERVAL` would have performed for the module M_{i+1} . This establishes that `INTERVAL` computes the same interval I_{i+1} with the same multiplicity as `TRUEINTERVAL` would have computed on M_{i+1} using the same order of exploration as the inductive hypothesis claims.

(“only if”): See the full version. ◀

► **Proposition 38.** $\text{INTERVAL}(\mathcal{F}, P)$ runs in $O(t^{\omega+2})$ time (proof in the full version).

4.3 Interval decomposability

Setup 3 still applies in §4.3. We consider the following problem.

► **Problem III.** Determine whether the module $M_{\mathcal{F}}$ is interval decomposable or not.

If the input module $M_{\mathcal{F}}$ is interval decomposable, then the algorithm INTERVAL computes all intervals in the barcode. However, if the module $M_{\mathcal{F}}$ is not interval decomposable, then the algorithm is not guaranteed to output all interval summands. We show that INTERVAL still can be used to solve Problem III. For this we test whether each of the output intervals I with multiplicity μ_I indeed supports a summand $N \cong \mathbb{I}_I^{\mu_I}$ of $M_{\mathcal{F}}$.

To do this we run Algorithm 3 in Asashiba et al. [1] for each of the output intervals of INTERVAL . Call this algorithm TESTINTERVAL which with an input interval I , returns $\mu_I > 0$ if the module $\mathbb{I}_I^{\mu_I}$ is a summand of M and 0 otherwise.

For each of the intervals I with multiplicity μ_I returned by $\text{INTERVAL}(\mathcal{F}, P)$ we test whether $\text{TESTINTERVAL}(I)$ returns a non-zero μ_I . The first time the test fails, we declare that $M_{\mathcal{F}}$ is not interval decomposable. This gives us a polynomial time algorithm (with complexity $O(t^{3\omega+2})$) to test whether a module induced by a given bifiltration is interval decomposable or not. It is a substantial improvement over the result of Asashiba et al. [1] who gave an algorithm for tackling the same problem. Their algorithm cleverly enumerates the intervals in the poset to test, but still tests exponentially many of them and hence may run in time that is exponential in t . Because of our algorithm INTERVAL , we can do the same test but only on polynomially many intervals.

■ **Algorithm 3** $\text{ISINTERVALDECOMP}(\mathcal{F}, P)$

- Step 1. $\mathcal{I} = \{(I_i, \mu_{I_i})\} \leftarrow \text{INTERVAL}(\mathcal{F}, P)$
 - Step 2. For every $I_i \in \mathcal{I}$ do
 - Step 2.1 $\mu \leftarrow \text{TESTINTERVAL}(M_{\mathcal{F}}, I_i)$
 - Step 2.2 If $\mu \neq \mu_{I_i}$ then output false; quit
 - Step 3. output true
-

► **Proposition 39.** $\text{ISINTERVALDECOMP}(\mathcal{F}, P)$ returns true if and only if $M_{\mathcal{F}}$ is interval decomposable. It takes $O(t^{3\omega+2})$ time.

Proof. By the contrapositive of Proposition 33, if for any of the computed interval(s) $I_i, i = 1, \dots, k$ by INTERVAL , $\mathbb{I}_{I_i}^{\mu_{I_i}}$ is not a summand of $M_{\mathcal{F}}$, then $M_{\mathcal{F}}$ is not interval decomposable. On the other hand, if every such interval module is a summand of $M_{\mathcal{F}}$, then we have that $M_{\mathcal{F}} \cong \bigoplus_{i=1}^k \mathbb{I}_{I_i}^{\mu_{I_i}}$ because $\dim(M_{\mathcal{F}})_p = \sum_i^k \dim(\mathbb{I}_{I_i}^{\mu_{I_i}})_p$ for every $p \in P$.

Time complexity : By Proposition 38, Step 1 runs in time $O(t^{\omega+2})$. We claim that $\dim(M_{\mathcal{F}}) = O(t^2)$ (see Proof of Proposition 38 in the full version). Therefore, INTERVAL returns at most $O(t^2)$ intervals. According to the analysis in Asashiba et al. [1], each test in Step 2.1 takes $O(((\dim M_{\mathcal{F}})^{\omega} + t)t^{\omega}) = O(t^{3\omega})$ time and thus $O(t^{3\omega+2})$ in total over all $O(t^2)$ tests which dominates the time complexity of ISINTERVALDECOMP . ◀

5 Discussion



The algorithm INTERVAL produces all intervals of an input interval decomposable module. What happens if the input module is not interval decomposable? We can show that the algorithm still produces intervals each supporting a submodule of an indecomposable of the input module (Figure 4), see [18] for details. Some other open questions that follow are: (i) Can we generalize Theorem 24 to d -parameter persistent homology for $d > 2$? (ii) Can the complexity of the algorithms be improved? (iii) In particular, can we improve the interval testing algorithm of Asashiba et al.?

References

- 1 Hideto Asashiba, Mickaël Buchet, Emerson G Escobar, Ken Nakashima, and Michio Yoshiwaki. On interval decomposability of 2d persistence modules. *arXiv preprint v2*, 2018. [arXiv:1812.05261](#).
- 2 Hideto Asashiba, Emerson G Escobar, Ken Nakashima, and Michio Yoshiwaki. On approximation of $2d$ -persistence modules by interval-decomposables. *arXiv preprint*, 2019. [arXiv:1911.01637](#).
- 3 Gorô Azumaya. Corrections and supplementaries to my paper concerning Krull-Remak-schmidt's theorem. *Nagoya Mathematical Journal*, 1:117–124, 1950.
- 4 Ulrich Bauer, Magnus B Botnan, Steffen Oppermann, and Johan Steen. Cotorsion torsion triples and the representation theory of filtered hierarchical clustering. *Advances in Mathematics*, 369:107171, 2020.
- 5 Leo Bettinauser, Peter Bubenik, and Parker B Edwards. Graded persistence diagrams and persistence landscapes. *Discrete & Computational Geometry*, pages 1–28, 2021.
- 6 Silvia Biasotti, Andrea Cerri, Patrizio Frosini, Daniela Giorgi, and Claudia Landi. Multidimensional size functions for shape comparison. *Journal of Mathematical Imaging and Vision*, 32(2):161–179, 2008.
- 7 Magnus Botnan, Vadim Lebovici, and Steve Oudot. On Rectangle-Decomposable 2-Parameter Persistence Modules. In *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164, pages 22:1–22:16, 2020.
- 8 Magnus Botnan and Michael Lesnick. Algebraic stability of zigzag persistence modules. *Algebraic & geometric topology*, 18(6):3133–3204, 2018.
- 9 Magnus Botnan, Steffen Oppermann, and Steve Oudot. Signed barcodes for multi-parameter persistence via rank decompositions and rank-exact resolutions. *arXiv preprint*, 2021. [arXiv:2107.06800](#).
- 10 Peter Bubenik and Alex Elchesen. Virtual persistence diagrams, signed measures, and wasserstein distance. *arXiv preprint*, 2020. [arXiv:2012.10514](#).
- 11 Chen Cai, Woojin Kim, Facundo Mémoli, and Yusu Wang. Elder-rule-staircodes for augmented metric spaces. *SIAM Journal on Applied Algebra and Geometry*, 5(3):417–454, 2021.
- 12 Gunnar Carlsson and Vin de Silva. Zigzag persistence. *Foundations of computational mathematics*, 10(4):367–405, 2010.
- 13 Gunnar Carlsson and Facundo Mémoli. Multiparameter hierarchical clustering methods. In *Classification as a Tool for Research*, pages 63–70. Springer, 2010.
- 14 Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, 2009.
- 15 Erin Chambers and David Letscher. Persistent homology over directed acyclic graphs. In *Research in Computational Topology*, pages 11–32. Springer, 2018.
- 16 Jérémy Cochoy and Steve Oudot. Decomposition of exact pfd persistence bimodules. *Discrete & Computational Geometry*, 63(2):255–293, 2020.
- 17 Tamal K. Dey and Tao Hou. Updating zigzag persistence and maintaining representatives over changing filtrations. *CoRR*, abs/2112.02352, 2021. [arXiv:2112.02352](#).

- 18 Tamal K. Dey, Woojin Kim, and Facundo Mémoli. Computing generalized rank invariant for 2-parameter persistence modules via zigzag persistence and its applications. *arXiv preprint*, 2021. [arXiv:2111.15058](https://arxiv.org/abs/2111.15058).
- 19 Tamal K. Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2022. URL: <https://www.cs.purdue.edu/homes/tamaldey/book/CTDAbook/CTDAbook.pdf>.
- 20 Tamal K. Dey and Cheng Xin. Generalized persistence algorithm for decomposing multiparameter persistence modules. *Journal of Applied and Computational Topology*, pages 1–52, 2022.
- 21 Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, January 2010.
- 22 Emerson G Escolar and Yasuaki Hiraoka. Persistence modules on commutative ladders of finite type. *Discrete & Computational Geometry*, 55(1):100–157, 2016.
- 23 Pierre Gabriel. Unzerlegbare darstellungen i. *Manuscripta Mathematica*, pages 71–103, 1972.
- 24 Michael Kerber. Multi-parameter persistent homology is practical. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020.
- 25 Woojin Kim and Facundo Mémoli. Rank invariant for zigzag modules. *arXiv preprint v1*, 2018. [arXiv:1810.11517](https://arxiv.org/abs/1810.11517).
- 26 Woojin Kim and Facundo Mémoli. Generalized persistence diagrams for persistence modules over posets. *Journal of Applied and Computational Topology*, 5(4):533–581, 2021.
- 27 Woojin Kim and Facundo Mémoli. Spatiotemporal persistent homology for dynamic metric spaces. *Discrete & Computational Geometry*, 66(3):831–875, 2021.
- 28 Woojin Kim and Samantha Moore. The generalized persistence diagram encodes the bigraded Betti numbers. *arXiv preprint*, 2021. [arXiv:2111.02551](https://arxiv.org/abs/2111.02551).
- 29 Michael Lesnick. *Multidimensional interleavings and applications to topological inference*. Stanford University, 2012.
- 30 Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015.
- 31 Michael Lesnick and Matthew Wright. Interactive visualization of 2-d persistence modules. *arXiv preprint*, 2015. [arXiv:1512.00180](https://arxiv.org/abs/1512.00180).
- 32 Alexander McCleary and Amit Patel. Edit distance and persistence diagrams over lattices. *arXiv preprint*, 2020. [arXiv:2010.07337](https://arxiv.org/abs/2010.07337).
- 33 Ezra Miller. Modules over posets: commutative and homological algebra. *arXiv preprint*, 2019. [arXiv:1908.09750](https://arxiv.org/abs/1908.09750).
- 34 Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the twenty-seventh Annual Symposium on Computational Geometry*, pages 216–225, 2011.
- 35 Amit Patel. Generalized persistence diagrams. *Journal of Applied and Computational Topology*, 1(3):397–419, 2018.
- 36 Gian-Carlo Rota. On the foundations of combinatorial theory i. theory of Möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 2(4):340–368, 1964.
- 37 Richard P Stanley. Enumerative combinatorics volume 1 second edition. *Cambridge studies in advanced mathematics*, 2011.

Tracking Dynamical Features via Continuation and Persistence

Tamal K. Dey   

Department of Computer Science, Purdue University, West Lafayette, IN, USA

Michał Lipiński   

Division of Computational Mathematics, Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland

Marian Mrozek   

Division of Computational Mathematics, Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland

Ryan Slechta   

Department of Computer Science, Purdue University, West Lafayette, IN, USA

Abstract

Multivector fields and combinatorial dynamical systems have recently become a subject of interest due to their potential for use in computational methods. In this paper, we develop a method to track an isolated invariant set – a salient feature of a combinatorial dynamical system – across a sequence of multivector fields. This goal is attained by placing the classical notion of the “continuation” of an isolated invariant set in the combinatorial setting. In particular, we give a “Tracking Protocol” that, when given a seed isolated invariant set, finds a canonical continuation of the seed across a sequence of multivector fields. In cases where it is not possible to continue, we show how to use zigzag persistence to track homological features associated with the isolated invariant sets. This construction permits viewing continuation as a special case of persistence.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Theory of computation → Computational geometry

Keywords and phrases combinatorial dynamical systems, continuation, index pair, Conley index, persistent homology

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.35

Related Version *Full Version*: <https://arxiv.org/abs/2203.05727>

Funding This work is partially supported by NSF grants CCF-2049010, CCF-1839252, Polish National Science Center under Maestro Grant 2014/14/A/ST1/00453, Opus Grant 2019/35/B/ST1/00874 and Preludium Grant 2018/29/N/ST1/00449

1 Introduction

Dynamical systems enter the field of data science in two ways: either directly, as in the case of dynamic data, or indirectly, as in the case of images, where gradient dynamics are useful. Forman’s discrete Morse theory [10, 11, 16] combines topology with gradient dynamics via *combinatorial vector fields*. Discrete Morse theory has been used to simplify datasets and to extract topological features from them [1, 7, 15, 25]. When coupled with persistent homology [7, 8, 9], this theory can be useful for analyzing complex data [12, 17, 18].

Conley theory [4] is a generalization of classical Morse theory beyond gradient dynamics. Conley’s approach to dynamical systems is motivated by the observation that in many areas, perhaps most notably biology, the differential equations governing systems of interest are known only roughly. Generally, this is due to the presence of several parameters which cannot be measured or estimated precisely. A similar situation occurs in data science, where



© Tamal K. Dey, Michał Lipiński, Marian Mrozek, and Ryan Slechta;
licensed under Creative Commons License CC-BY 4.0

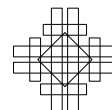
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 35; pp. 35:1–35:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



a time series dataset that is collected from a dynamical process only crudely approximates the underlying system. This observation has motivated recent studies [2, 6, 14, 20, 21, 23] on a variant of Conley theory for combinatorial vector fields.

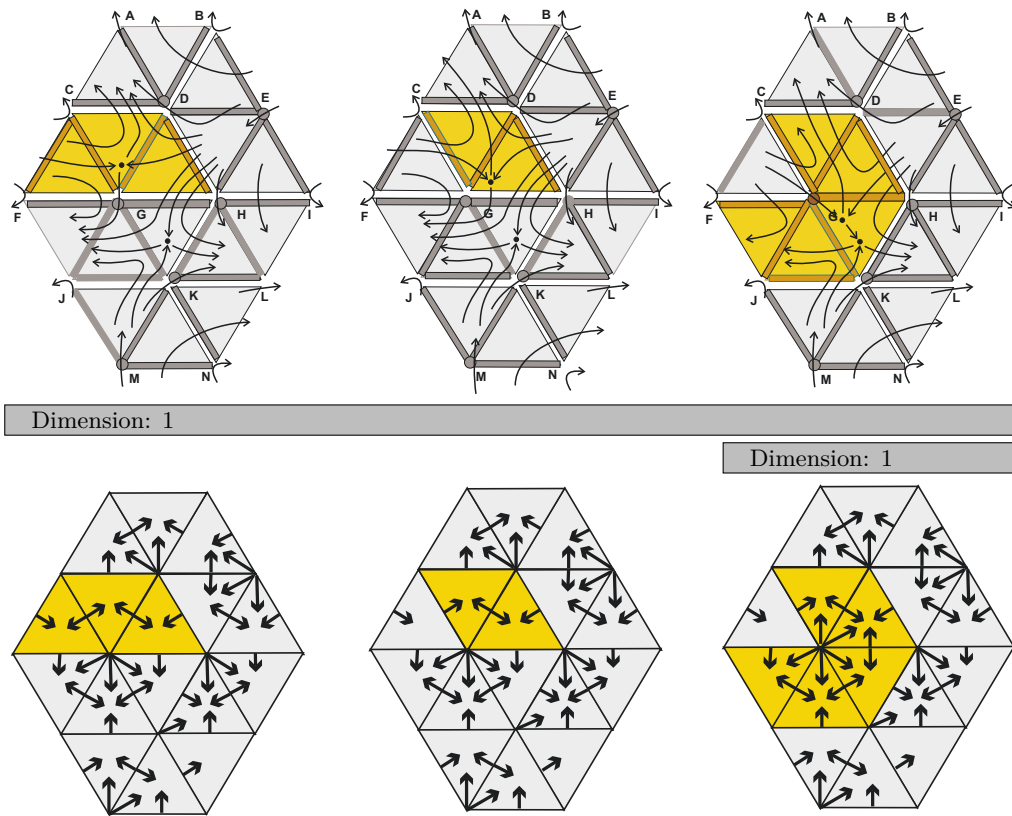
The primary objects of interest in Conley theory are *isolated invariant sets*, each of which is a salient feature of a vector field, together with an associated homological invariant called the *Conley index* (Section 2). Notably, isolated invariant sets with non-trivial Conley index persist under small perturbations. The geometry of the isolated invariant set may change, and even the topology may change, but the Conley index associated with the isolated invariant set remains the same. The isolated invariant set cannot suddenly vanish or change from an attractor to a repeller or vice versa. From this observation, we get the notion of the *continuation* of an isolated invariant set in a dynamical system to another one in a nearby system. This local idea becomes global by making the continuation relation transitive.

Given a path in the space of dynamical systems, one can track an invariant set along the path so long as the invariant set remains isolated. When the isolation is lost, continuation “breaks” and the Conley index is not well-defined. Typically, this may be observed when two isolated invariant sets merge. Isolation may eventually be regained, but there is no guarantee that the Conley index will be recovered. We propose to use persistence [7, 8, 9] in the discrete setting to connect continuations. First, we show how continuation can be detected and maintained algorithmically in a combinatorial multivector field, which is a discretized version of a continuous vector field. In fact, the continuation itself may be viewed as a special case of persistence where all bars persist for the duration of the continuation. When continuation “breaks,” we observe the birth and death of homology classes.

The combination of continuation and persistence allows us to algorithmically track an isolated invariant set and its associated Conley index in the setting of combinatorial dynamical systems. Recall that a combinatorial dynamical system is generated by a multivector field. A multivector field is a partition of a simplicial complex into sets that are convex with respect to the face poset. We track an isolated invariant set in a sequence of such fields where each field differs from its adjacent ones by an *atomic rearrangement*. Each *atomic rearrangement* is either an *atomic coarsening* or an *atomic refinement*. We show that an atomic refinement always permits continuation and thus the Conley index of the tracked invariant set persists. In the case of coarsening, we may not be able to continue. In such a case, we select an isolated invariant set that is a minimal perturbation of the previous one and compute the persistence of the Conley index between them. Hence, while there may come a point where we can no longer track an isolated invariant set, we can use persistence to track the lifetime of the homological features that are associated with the isolated invariant set.

The top row of Figure 1 presents flow lines from three flows on a simplicial complex with vertices marked from A to N. The same figure also shows three combinatorial multivector fields represented as three different partitions of the collection of cells into multivectors. Each multivector is depicted as a connected component, and it is easy to see that they are convex with respect to the face poset. The multivector fields are constructed as follows: if the flow transversely crosses an edge e into a triangle t , then e and t are put in the same multivector. Else, e is put into the same multivector as both of its incident triangles. If the flow line originating at a vertex v immediately enters triangle t , then v and t are put into the same multivector. See [22, 23] for additional information on this construction.

There are two saddle stationary points in each flow, indicated by small black dots. For all three flows, the lower saddle is located in triangle GHK . However, the upper one moves from triangle CDG in the left flow, through triangle DGH in the middle flow and finally it shares triangle GHK with the lower saddle in the right flow. On the combinatorial level, the



■ **Figure 1** (Top) Three multivector fields, corresponding to merging saddles, where the middle multivector field is an atomic refinement of the left and the right multivector field is an atomic coarsening of the middle. The persistence barcode associated with the isolated invariant sets – depicted in yellow – is shown in gray below the three figures. (Bottom) The multivector fields associated with the figure at the top using the standard multivector drawing convention.

upper saddle in the left flow is represented by an isolated invariant set S_1 consisting of one multivector $\{CFG, CDG, DGH, CF, CG, DG, DH\}$ marked in yellow. The Conley index of S_1 is non-trivial only in dimension one and has exactly one generator. Using methods from this paper, S_1 can be tracked to an isolated invariant set S_2 containing the upper saddle of the middle flow and consisting of one multivector $\{CDG, DGH, CG, DG, DH\}$, also marked in yellow. The isolated invariant set S_3 containing the upper saddle of the right flow consists of one multivector $\{CDG, DGH, FGJ, GJK, GHK, CG, DG, DH, FG, JK, GH, GJ, GK\}$, again marked in yellow. It is not a continuation of S_2 , because in the right flow the two saddles are too close to one another to be distinguishable with the resolution of the triangulation. Furthermore, the Conley index has changed. It is only nontrivial in dimension one, but unlike S_1 and S_2 , it has two generators. Hence, in the right multivector field, a new generator is born. We show how to capture the birth of this generator using persistence, and we depict the associated barcode beneath the top row of Figure 1. The familiar reader will note that the Conley index of S_3 is the same as that of a monkey saddle. However, because of the finite resolution, we cannot discriminate between two nearby saddles and a monkey saddle. One can view a multivector field as a combinatorial object that represents flows up to the resolution permitted by a triangulation. This purely combinatorial view of the top row of Figure 1 is presented in the bottom row. In subsequent examples, we use this style.

2 Combinatorial Dynamical Systems

In this section, we review multivector fields, combinatorial dynamical systems, and isolated invariant sets. Throughout this paper, K will always denote a finite simplicial complex. Furthermore, we will only consider simplicial homology [13, 24] with coefficients taken from a finite field. Much of the foundational work on combinatorial dynamical systems was first published in [21] and subsequently generalized in [20]. This work was heavily influenced by Forman’s discrete Morse theory [10, 11]. Combinatorial dynamical systems are constructed via multivector fields, which require a notion of convexity. Given a finite simplicial complex K , we let \leq denote the *face relation* on K . Formally, if $\sigma, \tau \in K$, then $\sigma \leq \tau$ if and only if σ is a face of τ . The set A is *convex* if for each pair $\sigma, \tau \in A$ where there exists a $\rho \in K$ satisfying $\sigma \leq \rho \leq \tau$, we have that $\rho \in A$.

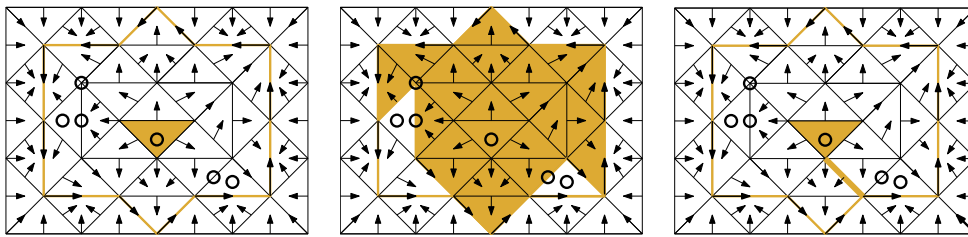
A *multivector* is a convex subset of a simplicial complex. A partition of K into multivectors is a *multivector field* on K . Multivectors are not required to have a unique maximal element under \leq , nor are they required to be connected. Disconnected multivectors do not appear in practice, and in the interest of legibility, all examples that we include in this paper only depict connected multivectors. However, all of our theoretical results do hold for disconnected multivectors. We draw a multivector V by drawing an arrow from each nonmaximal element $\sigma \in V$ to each maximal element $\tau \in V$ where $\sigma \leq \tau$. If σ is the only element of a multivector, or a *singleton*, then we mark σ with a circle. Each $\sigma \in K$ is contained in a unique multivector $V \in \mathcal{V}$. We denote the unique multivector in \mathcal{V} containing σ as $[\sigma]_{\mathcal{V}}$.

A multivector field \mathcal{V} induces dynamics on K . Given a simplex $\sigma \in K$, we denote the *closure* of σ as $\text{cl}(\sigma) := \{\tau \in K \mid \tau \leq \sigma\}$. For a set $A \subseteq K$, the closure of A is given by $\text{cl}(A) := \cup_{\sigma \in A} \text{cl}(\sigma)$. A set A is *closed* if and only if $A = \text{cl}(A)$. The multivector field \mathcal{V} induces a multivalued map $F_{\mathcal{V}} : K \multimap K$ where $F_{\mathcal{V}}(\sigma) := \text{cl}(\sigma) \cup [\sigma]_{\mathcal{V}}$. Informally, this will mean that if one is at a simplex σ , then one can move either to a face of σ or to a simplex τ in the same multivector as σ . We allow moving within any single multivector because, on the level of flows, the behavior within the multivector is beyond the resolution of the given simplicial complex. Conversely, we do not allow moving from a cell to its coface, unless they are in the same multivector, because this does not respect the underlying flow.

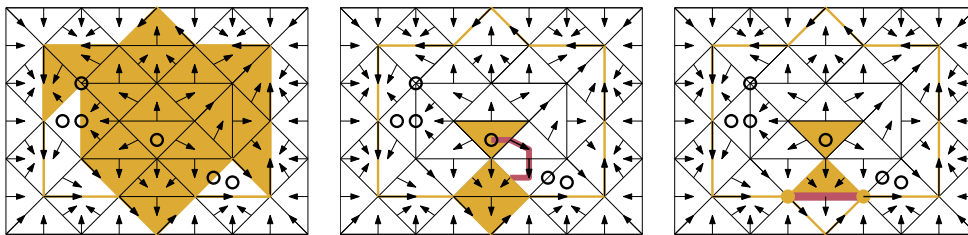
The multivalued map F gives a notion of *paths* and *solutions* to \mathcal{V} . We let $\mathbb{Z}_{[i,j]} := \mathbb{Z} \cap [i, j]$. A *path* is a function $\rho : \mathbb{Z}_{[0,n]} \rightarrow K$ where $\rho(i+1) \in F_{\mathcal{V}}(\rho(i))$ for $i \in \mathbb{Z}_{[0,n-1]}$. Likewise, a *solution* is a function $\rho : \mathbb{Z} \rightarrow K$ where $\rho(i+1) \in F_{\mathcal{V}}(\rho(i))$ for $i \in \mathbb{Z}$. However, there are several trivial solutions in a multivector field. If $\sigma \in K$, then there is a solution ρ where $\rho(i) = \sigma$ for all $i \in \mathbb{Z}$. That is, every simplex is a fixed point. This does not match the intuition from differential equations: only a very select set of simplices should be fixed points under $F_{\mathcal{V}}$. To enforce this, we use the notion of a *critical multivector*. But first, we define the *mouth* of a set A , denoted $\text{mo}(A)$, to be $\text{mo}(A) := \text{cl}(A) \setminus A$. The multivector V is *critical* if $H(\text{cl}(A), \text{mo}(A)) \neq 0$. Intuitively, critical multivectors with one maximal element correspond to stationary points in the flow setting. Thus, only simplices in critical multivectors should be fixed points under F . *Essential solutions* enforce this requirement [20].

► **Definition 1 (Essential Solution).** *Let $\rho : \mathbb{Z} \rightarrow K$ denote a solution to the multivector field \mathcal{V} . If for every $i \in \mathbb{Z}$ where $[\rho(i)]_{\mathcal{V}}$ is not critical, there exists a pair of integers $i^- < i < i^+$ where $[\rho(i^-)]_{\mathcal{V}} \neq [\rho(i)]_{\mathcal{V}}$ and $[\rho(i)]_{\mathcal{V}} \neq [\rho(i^+)]_{\mathcal{V}}$, then ρ is an essential solution.*

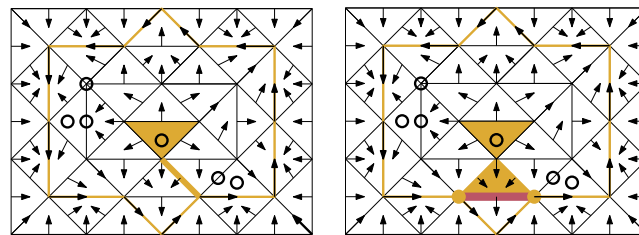
The *invariant part* of a set $A \subseteq K$, denoted $\text{Inv}_{\mathcal{V}}(A)$, is given by the set of simplices $\sigma \in A$ for which there exists an essential solution $\rho : \mathbb{Z} \rightarrow A$ where $\rho(i) = \sigma$ for some $i \in \mathbb{Z}$. A set $S \subseteq K$ is an *invariant set* if and only if $S = \text{Inv}_{\mathcal{V}}(S)$. For examples of invariant sets, see Figure 2. We are interested in a special type of invariant set.



■ **Figure 2** Three examples of an invariant set, marked in yellow.



■ **Figure 3** Three invariant sets on the same multivector field, marked in yellow. The left invariant set is isolated by the entire rectangle. The middle invariant set is isolated by its closure, but not by the rectangle because of the path in red. The right invariant set is isolated by neither its closure nor the rectangle, because there is a path from a yellow triangle, to the red edge, to the yellow vertex.



■ **Figure 4** Two invariant sets, marked in yellow, over the same multivector field. On the right, the invariant set includes the two yellow vertices marked with filled discs, but it excludes the red edge. The invariant set on the left is not \mathcal{V} -compatible, while the invariant set on the right is.

► **Definition 2** (Isolated Invariant Set, Isolating Set). *Let S be an invariant set under \mathcal{V} . If there exists a closed set N such that $F_{\mathcal{V}}(S) \subseteq N$ and every path $\rho : \mathbb{Z}_{[0,n]} \rightarrow N$ where $\rho(0), \rho(n) \in S$ has the property that $\rho(\mathbb{Z}_{[0,n]}) \subseteq S$, then S is isolated by N and S is an isolated invariant set. Moreover, the set N is an isolating set for S .*

Figure 3 illustrates the concept of isolation. An invariant set S is \mathcal{V} -compatible if S is equal to the union of a set of multivectors in \mathcal{V} . For examples, see Figure 4. This gives an equivalent formulation of an isolated invariant set.

► **Proposition 3** ([19], Proposition 4.1.21). *An invariant set S is isolated if and only if it is convex and \mathcal{V} -compatible.*

3 Tracking Isolated Invariant Sets

In this section, we introduce the protocol for tracking an isolated invariant set across multivector fields. Results in the continuous theory imply that under a sufficiently small perturbation, some homological features of an isolated invariant set do not change. Hence,

we require a notion of a small perturbation of a multivector field. In particular, let \mathcal{V} and \mathcal{V}' denote two multivector fields on K . If each multivector $V' \in \mathcal{V}'$ is contained in a multivector $V \in \mathcal{V}$, $|\mathcal{V}' \setminus \mathcal{V}| = 1$, and $|\mathcal{V} \setminus \mathcal{V}'| = 2$, then \mathcal{V}' is an *atomic refinement* of \mathcal{V} . It is so-called because \mathcal{V}' is obtained by “splitting” exactly one multivector in \mathcal{V} into two multivectors, while all the other multivectors remain the same. Symmetrically, we say that \mathcal{V} is an *atomic coarsening* of \mathcal{V}' . More broadly, it is said that \mathcal{V} and \mathcal{V}' are *atomic rearrangements* of each other. In Figures 5, 6, 7, 8, and 9, the two multivector fields are atomic rearrangements of each other. In these figures, we draw the multivectors that are splitting or merging in red.

Given an isolated invariant set S under \mathcal{V} , and an atomic rearrangement of \mathcal{V} denoted \mathcal{V}' , we aim to find an isolated invariant set S' that is a minimal perturbation of S . We accomplish this through two mechanisms: *continuation* and *persistence*. When we use continuation, or when we attempt to *continue*, we check if there exists an S' under \mathcal{V}' that is in some sense the same as S . If there is at least one such S' , then we choose a canonical one. This is explained in Section 4. If there is no S' to which we can continue, then we use persistence. In particular, we choose a canonical isolated invariant set S' under \mathcal{V}' , and while S does not continue to S' , we can use zigzag persistence to observe which features of S are absorbed by S' . We elaborate on this scheme in Section 5. To choose S' , we require the following result.

► **Proposition 4** ([19], Corollary 4.1.22). *Let A be a convex and \mathcal{V} -compatible set. Then $\text{Inv}_{\mathcal{V}}(A)$ is an isolated invariant set.*

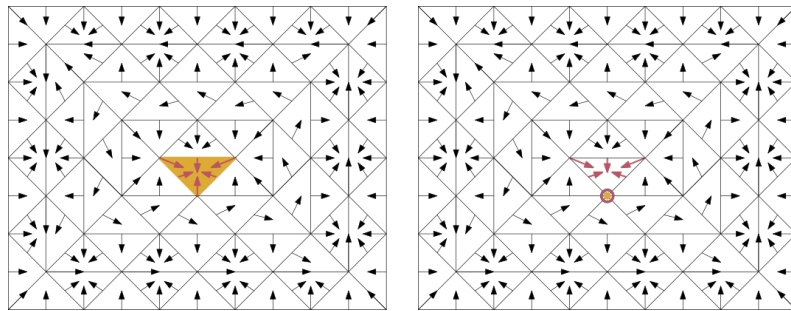
The set S is an isolated invariant set by assumption, so Proposition 3 implies that S is convex and \mathcal{V} -compatible. Thus, if S is also \mathcal{V}' -compatible, a natural choice is then to use Proposition 4 and take $S' := \text{Inv}_{\mathcal{V}'}(S)$. However, if S is not \mathcal{V}' -compatible, then the situation is more complicated. The set S is not \mathcal{V}' -compatible precisely when \mathcal{V}' is an atomic coarsening of \mathcal{V} , and the unique multivector $V \in \mathcal{V}' \setminus \mathcal{V}$, occasionally called the *merged multivector*, has the properties that $V \cap S \neq \emptyset$ and $V \not\subseteq S$. In such a case, we use the notation $\langle S \cup V \rangle_{\mathcal{V}'}$ to denote the intersection of all \mathcal{V}' -compatible and convex sets that contain $S \cup V$. The simplicial complex K is \mathcal{V}' -compatible and convex, so $\langle S \cup V \rangle_{\mathcal{V}'}$ always exists and $S \subsetneq \langle S \cup V \rangle_{\mathcal{V}'}$. We observe that $\langle S \cup V \rangle_{\mathcal{V}'}$ is \mathcal{V}' -compatible and convex, and thus it is the minimal convex and \mathcal{V}' -compatible set that contains S . In such a case, we use Proposition 4 and take $S' := \text{Inv}_{\mathcal{V}'}(\langle S \cup V \rangle_{\mathcal{V}'})$. These principles are enumerated in the following Tracking Protocol.

Tracking Protocol

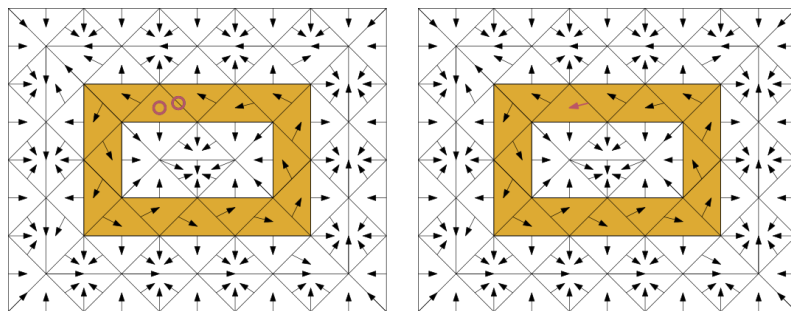
Given a nonempty isolated invariant set S under \mathcal{V} , and an atomic rearrangement of \mathcal{V} denoted \mathcal{V}' , use the following rules to find an isolated invariant set S' under \mathcal{V}' that corresponds to S .

1. Attempt to track via continuation:
 - a. If \mathcal{V}' is an atomic refinement of \mathcal{V} , then take $S' := \text{Inv}_{\mathcal{V}'}(S)$.
 - b. If \mathcal{V}' is an atomic coarsening of \mathcal{V} , and the unique merged multivector V has the property that $V \subseteq S$, then take $S' := \text{Inv}_{\mathcal{V}'}(S)$.
 - c. If \mathcal{V}' is an atomic coarsening of \mathcal{V} , and the unique merged multivector V has the property that $V \cap S = \emptyset$, then take $S' := \text{Inv}_{\mathcal{V}'}(S) = S$.
 - d. If \mathcal{V}' is an atomic coarsening of \mathcal{V} and the unique merged multivector V satisfies the formulae $V \cap S \neq \emptyset$ and $V \not\subseteq S$, then consider $A = \langle S \cup V \rangle_{\mathcal{V}'}$. If $\text{Inv}_{\mathcal{V}'}(A) = S$, then take $S' := \text{Inv}_{\mathcal{V}'}(A)$.
 - e. Else, it is impossible to track via continuation.
2. If it is impossible to track via continuation, then attempt to track via persistence:
 - f. If $A := \langle S \cup V \rangle_{\mathcal{V}'}$, then take $S' := \text{Inv}_{\mathcal{V}'}(A)$. If S and S' have a common isolating set, then use the technique in Equation 3 to find a zigzag filtration connecting them.
 - g. Otherwise, there is no natural choice of S' . See the full version for a possible strategy.

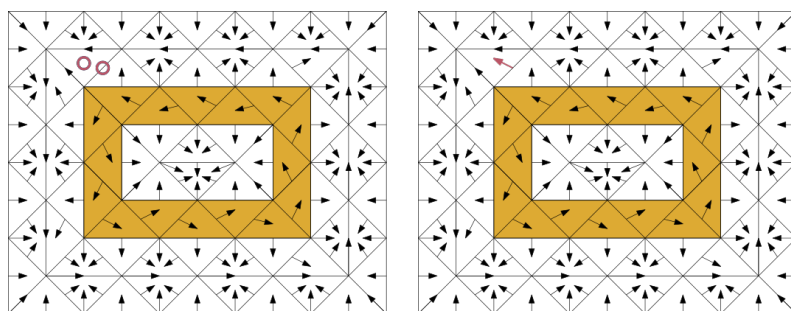
We include an example of Step 1a in Figure 5, Step 1b in Figure 6, Step 1c in Figure 7, Step 1d in Figure 8, and Step 2f in Figure 9. Each figure depicts a multivector field and a seed isolated invariant set on the left, and an atomic rearrangement and the resulting isolated invariant set on the right. By iteratively applying this protocol (until $S' = \emptyset$, in which case we are done), we can track how an isolated invariant set changes across several atomic rearrangements. See Figure 10 and the associated barcode in Figure 11. Any two multivector fields \mathcal{V}_1 and \mathcal{V}_2 can be related by a sequence of atomic rearrangements, and hence the Tracking Protocol can be used to track how an isolated invariant set changes across an arbitrary sequence of multivector fields. Additional details on this are in the full version.



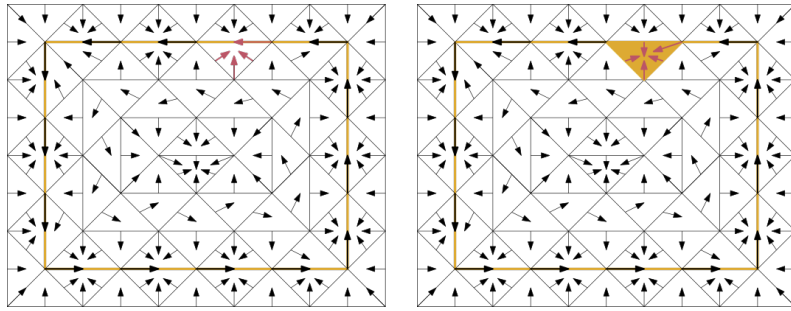
■ **Figure 5** Applying Step 1a to an invariant set (yellow, left) to get a new one (yellow, right).



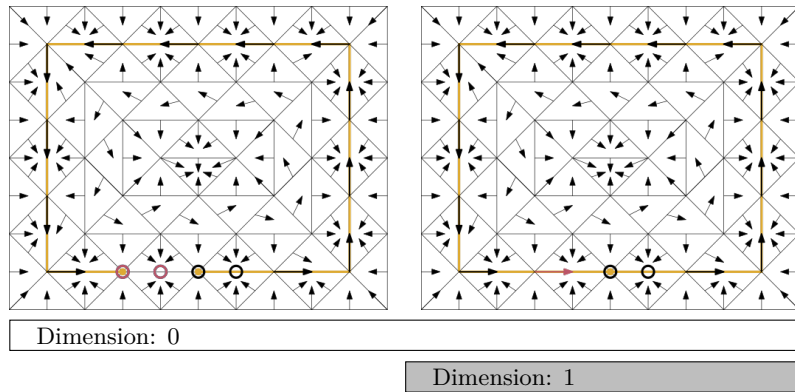
■ **Figure 6** Applying Step 1b to an invariant set (yellow, left) to get a new one (yellow, right).



■ **Figure 7** Applying Step 1c to an invariant set (yellow, left) to get a new one (yellow, right). The merged vector is outside of the invariant set on the left, so the invariant sets are the same.



■ **Figure 8** Applying Step 1d to an invariant set (yellow, left) to get a new one (yellow, right).



■ **Figure 9** Applying Step 2f to an invariant set (yellow, left) to get a new one (yellow, right). The associated persistence barcode is depicted below the figures.

4 Tracking via Continuation

Now, we introduce continuation in the combinatorial setting, and we justify the canonicity of the choices made in Step 1 of the Tracking Protocol. In addition, we show that if Step 1 is used to obtain S' from S , then S and S' are related by continuation. Continuation is closely related to the Conley index, so we begin with a brief review of the topic.

4.1 Index Pairs and the Conley Index

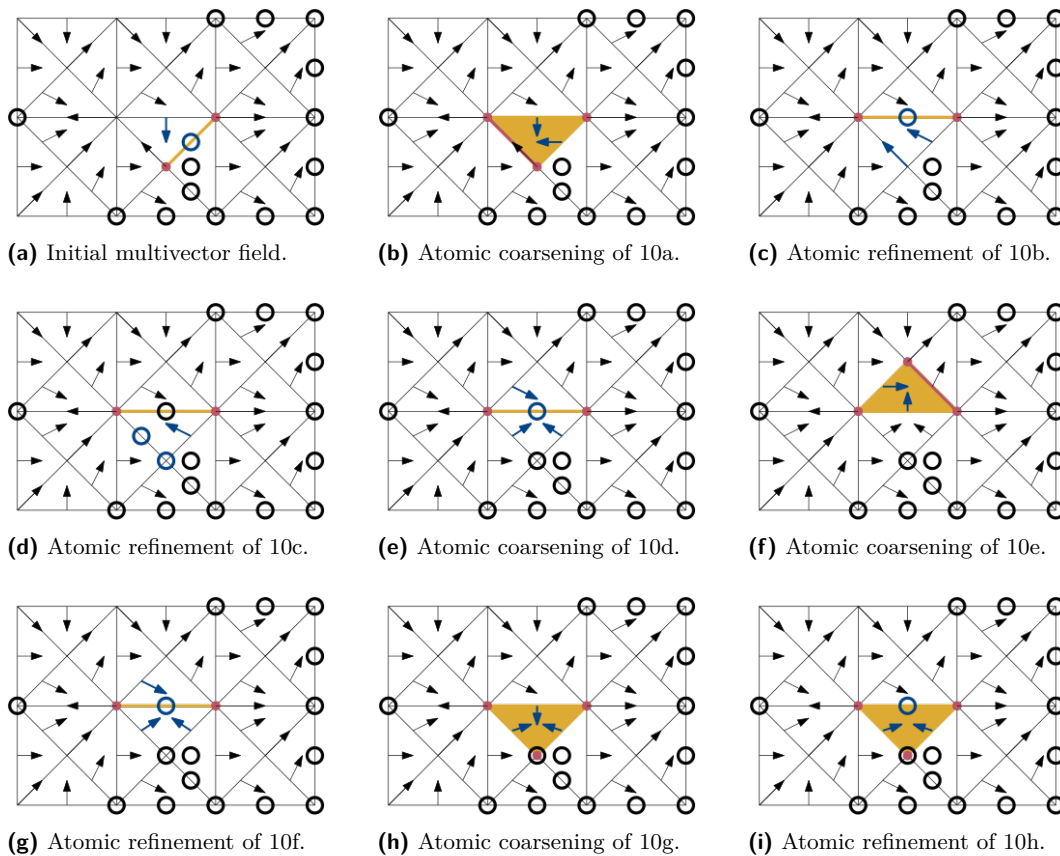
Originally developed in the classical setting by Conley [4], the Conley index associates a homological invariant with each isolated invariant set. It is defined through index pairs.

► **Definition 5 (Index Pair).** Let S denote an isolated invariant set under \mathcal{V} , and let P and E denote closed sets where $E \subseteq P$. If the following all hold, then (P, E) is an index pair for S :

1. $F_{\mathcal{V}}(P \setminus E) \subseteq P$
2. $F_{\mathcal{V}}(E) \cap P \subseteq E$
3. $S = \text{Inv}_{\mathcal{V}}(P \setminus E)$

For examples, see Figure 12. If (P, E) is an index pair for S , then the k -dimensional Conley index of S is $H_k(P, E)$. The authors in [20] showed that the Conley index is well-defined.

► **Theorem 6 ([20], Theorem 5.16).** Let (P, E) and (P', E') denote index pairs for S . Then $H_k(P, E) = H_k(P', E')$ for all $k \geq 0$.



■ **Figure 10** Subfigure 10a contains an initial multivector field and a seed isolated invariant set, which is a yellow edge. Each subsequent subfigure contains a multivector field that is an atomic refinement or atomic coarsening of the previous. The isolated invariant set that we get by iteratively applying the Tracking Protocol is depicted in yellow. Splitting and merging multivectors are in blue.

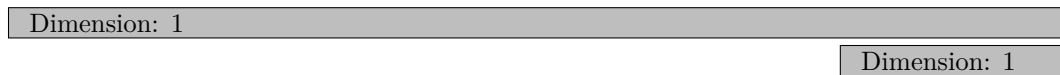
For a single isolated invariant set, there may be many possible index pairs. However, we can choose a canonical one, namely, the minimal index pair.

► **Proposition 7** ([20], Proposition 5.3). *Let S denote an isolated invariant set. The pair $(cl(S), mo(S))$ is an index pair for S .*

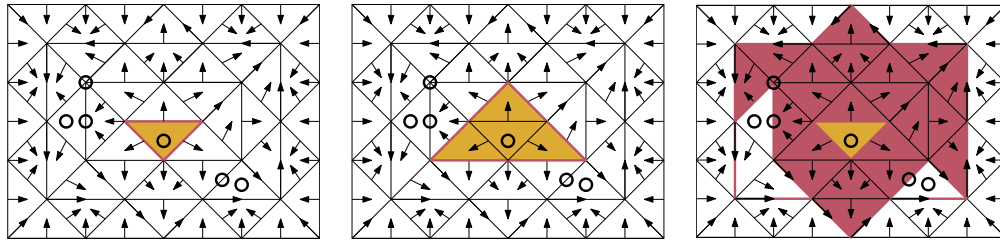
The following two propositions show that convex and \mathcal{V} -compatible sets are crucial for finding index pairs.

► **Proposition 8** ([20], Proposition 5.6). *Let (P, E) be an index pair under \mathcal{V} . Then $P \setminus E$ is convex and \mathcal{V} -compatible.*

► **Proposition 9.** *If A is convex and \mathcal{V} -compatible, then $(cl(A), mo(A))$ is an index pair for $Inv_{\mathcal{V}}(A)$.*



■ **Figure 11** The barcode associated with the tracked invariant sets in Figure 10. Starting with subfigure 10h, we see the birth of a new 1-dimensional homology generator.



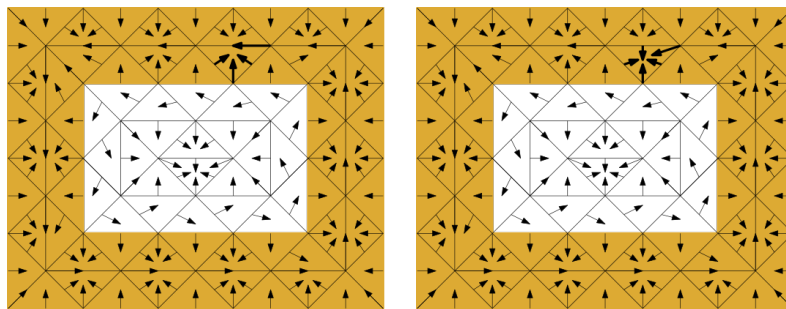
■ **Figure 12** All three images depict an index pair for the yellow triangle marked with a black circle. P is given by the red and yellow simplices, while E is given by the red simplices.

Proof. By Proposition 4 the set $S = \text{Inv}_{\mathcal{V}}(A)$ is an isolated invariant set. Since $\text{cl}(A) \setminus \text{mo}(A) = A$, we immediately get condition 3 from Definition 5. Since A is \mathcal{V} -compatible we get $F_{\mathcal{V}}(A) = \text{cl} A$, and thus, condition 1. To see condition 2 consider $x \in F_{\mathcal{V}}(\text{mo}(A))$. By the definition of $F_{\mathcal{V}}$ there exists an $a \in \text{mo}(A)$ such that either $x \in [a]_{\mathcal{V}}$ or $x \in \text{cl}(a)$. In the first case $x \notin A$, because A is \mathcal{V} -compatible and $a \notin A$. Therefore $[a]_{\mathcal{V}} \cap \text{cl}(A) \subseteq \text{cl}(A) \setminus A = \text{mo}(A)$. If $x \in \text{cl}(a)$ then $x \in \text{mo} A$, because $\text{mo}(A)$ is closed. Hence, it follows that $F_{\mathcal{V}}(\text{mo}(A)) \cap \text{cl}(A) \subseteq \text{mo}(A)$. ◀

4.2 Combinatorial Continuation and the Tracking Protocol

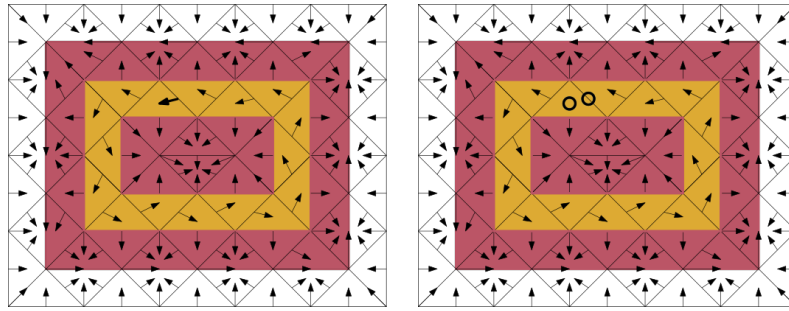
We now move to placing continuation in the combinatorial setting and explaining Step 1 of the Tracking Protocol. In essence, a continuation captures the presence of the “same” isolated invariant set across multiple multivector fields. We then show that Step 1 of the Tracking Protocol does use continuation to track an isolated invariant set.

► **Definition 10.** Let S_1, S_2, \dots, S_n denote a sequence of isolated invariant sets under the multivector fields $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_n$, where each \mathcal{V}_i is defined on a fixed simplicial complex K . We say that isolated invariant set S_1 continues to isolated invariant set S_n whenever there exists a sequence of index pairs $(P_1, E_1), (P_2, E_2), \dots, (P_{n-1}, E_{n-1})$ where (P_i, E_i) is an index pair for both S_i and S_{i+1} . Such a sequence is a sequence of connecting index pairs.



■ **Figure 13** An index pair, where P is in yellow and E is empty, for the isolated invariant sets in Figure 8. There is a common index pair for both isolated invariant sets, so they form a continuation.

Each index pair (P_i, E_i) in a connecting sequence of index pairs is an index pair for a pair of consecutive isolated invariant sets S_i and S_{i+1} (see Figures 13 and 14). Hence, the isolated invariant sets in the continuation all have the same Conley index. In this sense, we are capturing the “same” isolated invariant set. In Step 1 of the Tracking Protocol, we first



■ **Figure 14** An index pair, where P is given by the yellow and red simplices and E is given by the red simplices, for the isolated invariant sets in Figure 6. Thus, they form a continuation.

attempt to track the isolated invariant set S via continuation. That is, if we use Step 1, then we choose S' such that S and S' have a common index pair, say (P, E) . It so happens that (P, E) is easy to find algorithmically. We begin with the refinement case, or Step 1a.

► **Theorem 11.** *Let \mathcal{V} and \mathcal{V}' denote multivector fields where \mathcal{V}' is an atomic refinement of \mathcal{V} . Let A be a \mathcal{V} -compatible and convex set. The pair $(\text{cl}(A), \text{mo}(A))$ is an index pair for both $\text{Inv}_{\mathcal{V}}(A)$ under \mathcal{V} and $\text{Inv}_{\mathcal{V}'}(A)$ under \mathcal{V}' .*

The proof of Theorem 11 is included in the full version. In Step 1a of the Tracking Protocol, where \mathcal{V}' is an atomic refinement of \mathcal{V} , we choose $S' := \text{Inv}_{\mathcal{V}'}(S)$. By Proposition 3, it follows that S is \mathcal{V} -compatible. By identical reasoning to that presented in the proof of Theorem 11, it follows that S is also \mathcal{V}' -compatible. Hence, Theorem 11 implies that $(\text{cl}(S), \text{mo}(S))$ is an index pair for both $S = \text{Inv}_{\mathcal{V}}(S)$ and $S' = \text{Inv}_{\mathcal{V}'}(S)$. Thus, S and S' share an index pair.

The case of an atomic coarsening, corresponding to Steps 1b, 1c, and 1d of the Tracking Protocol, is more complicated. Recall that if \mathcal{V}' is an atomic coarsening of \mathcal{V} , then the unique multivector $V \in \mathcal{V}' \setminus \mathcal{V}$ is called the *merged multivector*.

► **Theorem 12.** *Let \mathcal{V} and \mathcal{V}' denote multivector fields where \mathcal{V}' is an atomic coarsening of \mathcal{V} . Let A be a convex and \mathcal{V} -compatible set, and let $V \in \mathcal{V}'$ be the unique merged multivector. If $V \subseteq A$ or $V \cap A = \emptyset$, then $(\text{cl}(A), \text{mo}(A))$ is an index pair for both $\text{Inv}_{\mathcal{V}}(A)$ and $\text{Inv}_{\mathcal{V}'}(A)$.*

Proof. If $V \cap A = \emptyset$, then A is both \mathcal{V} -compatible and \mathcal{V}' -compatible. Thus, Proposition 9 implies that $(\text{cl}(A), \text{mo}(A))$ is an index pair for both $S = \text{Inv}_{\mathcal{V}}(A)$ and $S' = \text{Inv}_{\mathcal{V}'}(A)$.

If $V \subseteq A$, then by the same reasoning as in the proof of Theorem 11, it follows that A is both \mathcal{V} -compatible and \mathcal{V}' -compatible. Thus, Proposition 9 implies that $(\text{cl}(A), \text{mo}(A))$ is an index pair for both $\text{Inv}_{\mathcal{V}}(A)$ and $\text{Inv}_{\mathcal{V}'}(A)$. ◀

By Proposition 3, S is convex and \mathcal{V} -compatible. Theorem 12 implies that if $V \subseteq S$ or $V \cap S = \emptyset$, then $(\text{cl}(S), \text{mo}(S))$ is an index pair for both $\text{Inv}_{\mathcal{V}}(S) = S$ and $\text{Inv}_{\mathcal{V}'}(S) = S'$. In Steps 1b and 1c of the Tracking Protocol, S' is chosen as $\text{Inv}_{\mathcal{V}'}(S)$. Hence, the index pair $(\text{cl}(S), \text{mo}(S))$ is an index pair for both S and S' .

A more complicated case is Step 1d, where $V \cap S \neq \emptyset$ and $V \not\subseteq S$. Recall that $A := \langle S \cup V \rangle_{\mathcal{V}'}$ denotes the intersection of all convex and \mathcal{V}' -compatible sets that contain $S \cup V$, and in particular, A is convex and \mathcal{V}' -compatible. In Step 1d of the Tracking Protocol, we first check if $S = \text{Inv}_{\mathcal{V}}(A)$. By Proposition 9, if $S = \text{Inv}_{\mathcal{V}}(A)$, then $(\text{cl}(A), \text{mo}(A))$ is an index pair for S . The set $\langle S \cup V \rangle_{\mathcal{V}'}$ is necessarily \mathcal{V} -compatible, because it is \mathcal{V}' -compatible by

35:12 Tracking via Continuation and Persistence

construction and it contains the unique merged multivector. Hence, Proposition 4 implies that $S' := \text{Inv}_{\mathcal{V}'}(A)$ is an isolated invariant set. Thus, Proposition 9 implies that $(\text{cl}(A), \text{mo}(A))$ is also an index pair for S' . Hence, if Step 1d gives S' , there is an index pair for S and S' .

In Step 1e of the Tracking Protocol, we claim that if $S \neq \text{Inv}_{\mathcal{V}}(A)$, then it is not possible to continue. Equivalently, there is no S' that shares an index pair with S .

► **Theorem 13.** *Let S denote an isolated invariant set under \mathcal{V} and let \mathcal{V}' denote an atomic coarsening of \mathcal{V} where the unique merged multivector $V \in \mathcal{V}' \setminus \mathcal{V}$ satisfies the formulae $V \cap S \neq \emptyset$ and $V \not\subseteq S$. Furthermore, let $A := \langle S \cup V \rangle_{\mathcal{V}'}$. If $S \neq \text{Inv}_{\mathcal{V}}(A)$, then there does not exist an isolated invariant set S' under \mathcal{V}' for which there is an index pair (P, E) satisfying $\text{Inv}_{\mathcal{V}}(P \setminus E) = S$ and $\text{Inv}_{\mathcal{V}'}(P \setminus E) = S'$.*

Proof. Suppose that $S \neq \text{Inv}_{\mathcal{V}}(A)$ and there exists an index pair, (P, E) , for both S under \mathcal{V} and some S' under \mathcal{V}' . By Proposition 8, the set $P \setminus E$ must be convex and \mathcal{V}' -compatible. Since $S \subseteq P \setminus E$ and A is the smallest convex and \mathcal{V}' -compatible set containing S , it follows that $A \subseteq P \setminus E$. Hence, $\text{Inv}_{\mathcal{V}}(A) \subseteq \text{Inv}_{\mathcal{V}}(P \setminus E)$. By assumption, $S \subsetneq \text{Inv}_{\mathcal{V}}(A)$. Thus, $S \subsetneq \text{Inv}_{\mathcal{V}}(P \setminus E)$. This implies that (P, E) is not an index pair for S , a contradiction. ◀

4.3 Characterizing Tracked Isolated Invariant Sets

Step 1 of the Tracking Protocol provides an avenue for tracking an isolated invariant set across a sequence of atomic rearrangements. In this subsection, we justify the canonicity of the selected isolated invariant set in Step 1 of the Tracking Protocol. First, we observe that we always have an inclusion. We prove Theorem 14 in the full version.

► **Theorem 14.** *If S' is obtained by applying Step 1 of the Tracking Protocol to S , then we have $S \subseteq S'$ or $S' \subseteq S$.*

Furthermore, isolated invariant sets chosen by Step 1 minimize the perturbation to S in terms of the number of inclusions. We include the proof for Proposition 15 in the full version.

► **Proposition 15.** *Let S be an isolated invariant set under \mathcal{V} , and let S' be an isolated invariant set under \mathcal{V}' that is obtained by applying Step 1 of the Tracking Protocol to S . If S'' is any isolated invariant set under \mathcal{V}' that shares a common index pair with S , then $S' \subseteq S''$. Moreover, if $S'' \subseteq S$, then $S' = S''$.*

5 Tracking via Persistence

In the previous section, we explicated Step 1 of the protocol, which uses continuation to track an isolated invariant set across a changing multivector field. In this section, we first place continuation in the persistence framework by showing how to translate the idea of combinatorial continuation into a zigzag filtration [3, 7] that does not introduce spurious information. Then, we use the persistence view of continuation to justify Step 2f of the Tracking Protocol, which permits us to capture changes in an isolated invariant set when no continuation is possible. In particular, it permits us to track an isolated invariant set even in the presence of a bifurcation that changes the Conley index. If the isolated invariant set that we are tracking collides, or *merges*, with another isolated invariant set, then we follow the newly formed isolated invariant set, and persistence captures which aspects of our original isolated invariant set persist into the new one. Conversely, if an isolated invariant set splits, we track the smallest isolated invariant set that contains all of the child invariant sets. We begin by reviewing some results on computing the persistence of the Conley index from [5].

5.1 Conley Index Persistence

In [5], the authors were interested in computing the changing Conley index across a sequence of isolated invariant sets. A naive approach to computing the persistence of the Conley index is, if given two index pairs (P_1, E_1) and (P_2, E_2) , to take the intersection of the index pairs to obtain the zigzag filtration $(P_1, E_1) \supseteq (P_1 \cap P_2, E_1 \cap E_2) \subseteq (P_2, E_2)$. However, the intersection of index pairs is generally not an index pair, and as a consequence, the barcode associated with this zigzag filtration does not capture a changing Conley index. In addition, due to the fact that $(P_1 \cap P_2, E_1 \cap E_2)$ need not be an index pair, the barcode is frequently erratic. An example is in Figure 15. To solve this issue, we consider *index pairs in N* [5].

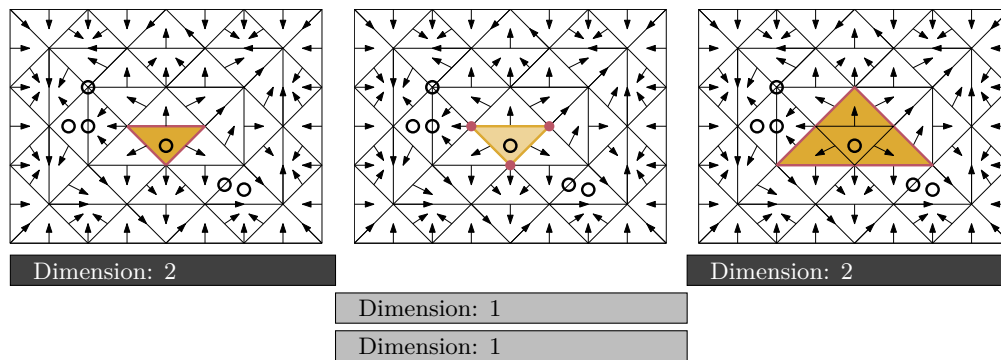


Figure 15 All three images depict the same multivector field, which includes a yellow repelling fixed point (triangle, marked with a black circle). (left) and (right) depict two different index pairs, (P_l, E_l) and (P_r, E_r) , for the repelling fixed point: P_l and P_r consist of yellow and red simplices and E_l and E_r consist of red simplices. The intersection $(P_l \cap P_r, E_l \cap E_r)$ is depicted in the middle. Check that this pair is not an index pair because if e denotes a yellow edge, then $F_{\mathcal{V}}(e) \not\subseteq P_l \cap P_r$. Beneath, we depict the barcode that is associated with the zigzag filtration $(P_l, E_l) \supseteq (P_l \cap P_r, E_l \cap E_r) \subseteq (P_r, E_r)$. Because (P_l, E_l) and (P_r, E_r) are both index pairs for the same repelling fixed point, we would expect the barcode to be full. However, as $(P_l \cap P_r, E_l \cap E_r)$ is not an index pair for the repelling fixed point, its relative homology can change drastically.

► **Definition 16.** Let S denote an isolated invariant set, and let N denote an isolating set for S . The pair of closed sets (P, E) is an index pair for S in N if all of the following hold:

1. $F_{\mathcal{V}}(P \setminus E) \subseteq N$
2. $F_{\mathcal{V}}(E) \cap N \subseteq E$
3. $F_{\mathcal{V}}(P) \cap N \subseteq P$
4. $S = \text{Inv}_{\mathcal{V}}(P \setminus E)$

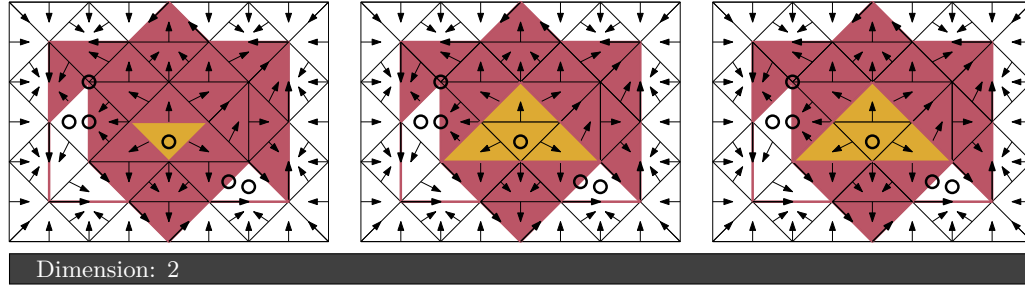
Every index pair in N is also an index pair in the sense of Definition 5 (see [5]). The canonical choice of an index pair for S can be used to obtain a canonical index pair for S in N via the *push forward*. The push forward of a set A in N , denoted $\text{pf}_{\mathcal{V}}(A, N)$, is given by the set of simplices $\sigma \in N$ for which there exists a path $\rho : \mathbb{Z}_{[0,n]} \rightarrow N$ in \mathcal{V} where $\rho(0) \in A$ and $\rho(n) = \sigma$. If \mathcal{V} is clear from the context we write $\text{pf}(A, N)$.

► **Theorem 17** ([5], Theorem 15). Let S be an isolated invariant set under \mathcal{V} , and let N be an isolating set for S . The pair $(\text{pf}_{\mathcal{V}}(\text{cl}(S), N), \text{pf}_{\mathcal{V}}(\text{mo}(S), N))$ is an index pair in N for S .

Index pairs in N are particularly useful because, unlike standard index pairs, their intersection is guaranteed to be an index pair. For two multivector fields \mathcal{V}_1 and \mathcal{V}_2 , an intermediate multivector field is $\mathcal{V}_1 \bar{\cap} \mathcal{V}_2$, where $\mathcal{V}_1 \bar{\cap} \mathcal{V}_2 := \{V_1 \cap V_2 \mid V_1 \in \mathcal{V}_1, V_2 \in \mathcal{V}_2\}$.

► **Theorem 18** ([5], Theorem 10). *Let (P_1, E_1) and (P_2, E_2) denote index pairs in N under \mathcal{V}_1 and \mathcal{V}_2 , respectively. The pair $(P_1 \cap P_2, E_1 \cap E_2)$ is an index pair in N under $\mathcal{V}_1 \cap \mathcal{V}_2$.*

Hence, given an index pair (P_1, E_1) in N under \mathcal{V}_1 and an index pair (P_2, E_2) in N under \mathcal{V}_2 , we can obtain a relative zigzag filtration where each pair is an index pair under a different multivector field. This zigzag filtration permits capturing a changing Conley index via persistence. We include an example in Figure 16.



■ **Figure 16** All three images depict the same multivector field in Figure 15. The left and the right images depict an index pair in N , where N is the entire rectangle. The color convention is the same as in Figure 15: red and yellow simplices are in P , and red simplices are in E . Unlike Figure 15, Theorem 18 implies that the intersected pair in the middle is an index pair. The persistence barcode, capturing the static Conley index, is depicted below the three images.

5.2 From continuation to filtration

Now, we show that a continuation of an isolated invariant set S_1 to S_{n+1} can be expressed in terms of persistence. Namely, a corresponding sequence of connecting index pairs (P_1, E_1) , (P_2, E_2) , \dots , (P_n, E_n) can be turned into a *zigzag filtration*, that is a sequence of pairs $\{(A_i, B_i)\}_{i=1}^m$ such that either $(A_i, B_i) \subseteq (A_{i+1}, B_{i+1})$ or $(A_{i+1}, B_{i+1}) \subseteq (A_i, B_i)$. Ideally, each (A_i, B_i) would be an index pair for some S_j from the initial continuation so as to not introduce spurious invariant sets or Conley indices. A connecting index pair (P_i, E_i) is an index pair for both S_i under \mathcal{V}_i and for S_{i+1} under \mathcal{V}_{i+1} . Thus, (P_i, E_i) and (P_{i+1}, E_{i+1}) are both index pairs for S_{i+1} under \mathcal{V}_{i+1} . We will construct auxiliary index pairs for S_{i+1} and then relate (P_i, E_i) and (P_{i+1}, E_{i+1}) with a zigzag filtration using these auxiliary pairs. If we can connect all adjacent pairs (P_i, E_i) and (P_{i+1}, E_{i+1}) with a zigzag filtration, then we can concatenate all of these zigzag filtrations and transform a sequence of connecting index pairs into a larger zigzag filtration. The following results are important for achieving this.

► **Proposition 19** ([20], Proposition 5.2). *Let (P, E) denote an index pair for S . The set P is an isolating set for S .*

► **Proposition 20.** *Let (P, E) denote an index pair for S under \mathcal{V} . The pair (P, E) is an index pair for S in P under \mathcal{V} .*

Proof. First, we observe that $S = \text{Inv}_{\mathcal{V}}(P \setminus E)$ because (P, E) is an index pair. In addition, $F_{\mathcal{V}}(P) \cap N = F_{\mathcal{V}}(P) \cap P \subseteq P$ by definition. Since (P, E) is an index pair, it has the property that $F_{\mathcal{V}}(P \setminus E) \subseteq P$. In the case of index pairs in N , we require that $F_{\mathcal{V}}(P \setminus E) \subseteq N = P$, so this case is immediately satisfied. Finally, because (P, E) is an index pair, $F_{\mathcal{V}}(E) \cap P \subseteq E$. Thus, $F_{\mathcal{V}}(E) \cap N = F_{\mathcal{V}}(E) \cap P \subseteq E$. ◀

► **Theorem 21.** *Let (P_1, E_1) and (P_2, E_2) denote index pairs for S in N under \mathcal{V} . The pair $(P_1 \cap P_2, E_1 \cap E_2)$ is an index pair for S in N under \mathcal{V} .*

We include the proof of Theorem 21 in the full version.

Now, we move to using these results to translate a sequence of connecting index pairs $\{(P_i, E_i)\}_{i=1}^n$ into a zigzag filtration. For $1 < i \leq n$, (P_{i-1}, E_{i-1}) and (P_i, E_i) are both index pairs for S_i . By Proposition 7, the pair $(\text{cl}(S_i), \text{mo}(S_i))$ is an index pair for S_i . Hence, a natural approach is to find a zigzag filtration that connects (P_i, E_i) with $(\text{cl}(S_i), \text{mo}(S_i))$ and a zigzag filtration that connects (P_{i-1}, E_{i-1}) with $(\text{cl}(S_i), \text{mo}(S_i))$. If we can find such zigzag filtrations for all S_i , then we can concatenate all of them and obtain a zigzag filtration that connects (P_1, E_1) with (P_n, E_n) . We depict the resulting zigzag filtration in Equation 1.

$$(P_1, E_1) \supseteq \dots \supseteq (\text{cl}(S_2), \text{mo}(S_2)) \subseteq \dots \subseteq (P_2, E_2) \supseteq \dots \supseteq (\text{cl}(S_3), \text{mo}(S_3)) \subseteq \dots (P_n, E_n) \tag{1}$$

We connect $(\text{cl}(S_i), \text{mo}(S_i))$ with (P_i, E_i) , and (P_{i-1}, E_{i-1}) connects with $(\text{cl}(S_i), \text{mo}(S_i))$ symmetrically. By Proposition 19, P_i is an isolating set for S_i . Thus, by Theorem 17, $(\text{pf}_{\mathcal{V}_i}(\text{cl}(S_i), P_i), \text{pf}_{\mathcal{V}_i}(\text{mo}(S_i), P_i))$ is an index pair for S_i in P_i . Proposition 20 implies that (P_i, E_i) is an index pair for S_i in P_i . By Theorem 21, $(P_i \cap \text{pf}_{\mathcal{V}_i}(\text{cl}(S_i), P_i), E_i \cap \text{pf}_{\mathcal{V}_i}(\text{mo}(S_i), P_i))$ is an index pair for S_i in P_i . Hence, we get the following zigzag filtration:

$$\begin{aligned} (\text{cl}(S_i), \text{mo}(S_i)) \subseteq (\text{pf}_{\mathcal{V}_i}(\text{cl}(S_i), P_i), \text{pf}_{\mathcal{V}_i}(\text{mo}(S_i), P_i)) \supseteq \\ (P_i \cap \text{pf}_{\mathcal{V}_i}(\text{cl}(S_i), P_i), E_i \cap \text{pf}_{\mathcal{V}_i}(\text{mo}(S_i), P_i)) \subseteq (P_i, E_i) \end{aligned} \tag{2}$$

Every pair in Equation 2 is an index pair for S_i under \mathcal{V}_i . Thus, we do not introduce any spurious invariant sets. We can concatenate these filtrations to get Equation 1.

We now analyze the barcode obtained for 1. we prove Theorem 22 in the full version.

► **Theorem 22.** *For every $k \geq 0$, the k -dimensional barcode of a connecting sequence of index pairs $\{(P_i, E_i)\}_{i=1}^n$ has m bars $[1, n]$ if $\dim H_k(P_1, E_1) = m$.*

5.3 Tracking beyond continuation

In the previous subsection, we showed how to convert a connecting sequence of index pairs into a zigzag filtration. Furthermore, we observed that it produces “full” barcodes - they have one bar for each basis element of the Conley index that persists for the length of the filtration. This change of perspective allows us to generalize our protocol to handle cases when it is impossible to continue.

In particular, we consider Step 2f of the protocol. Let S denote an isolated invariant set under \mathcal{V} , and \mathcal{V}' is an atomic coarsening of \mathcal{V} where the merged multivector V has the property that $V \cap S \neq \emptyset$ and $V \not\subseteq S$. In such a case, we consider $A := \langle S \cup V \rangle_{\mathcal{V}'}$ and take $S' = \text{Inv}_{\mathcal{V}'}(A)$. Theorem 13 implies that if $S \neq \text{Inv}_{\mathcal{V}}(A)$, then it is impossible to continue. However, it may be possible to compute persistence in a way that resembles continuation. Let $B := \text{cl}(S) \cup \text{cl}(S')$. Trivially, B is closed. If B is an isolating set for both S and S' , then we say that S and S' are *adjacent*. By Theorem 17, $(\text{pf}_{\mathcal{V}}(\text{cl}(S), B), \text{pf}_{\mathcal{V}}(\text{mo}(S), B))$ is an index pair for S in B . Similarly, $(\text{pf}_{\mathcal{V}'}(\text{cl}(S'), B), \text{pf}_{\mathcal{V}'}(\text{mo}(S'), B))$ is an index pair for S' in B . Thus, we can use Theorem 18 to obtain the following zigzag filtration.

$$\begin{aligned} (\text{cl}(S), \text{mo}(S)) \subseteq (\text{pf}_{\mathcal{V}}(\text{cl}(S), B), \text{pf}_{\mathcal{V}}(\text{mo}(S), B)) \\ \supseteq (\text{pf}_{\mathcal{V}}(\text{cl}(S), B) \cap \text{pf}_{\mathcal{V}'}(\text{cl}(S'), B), \text{pf}_{\mathcal{V}}(\text{mo}(S), B) \cap \text{pf}_{\mathcal{V}'}(\text{mo}(S'), B)) \subseteq \\ (\text{pf}_{\mathcal{V}'}(\text{cl}(S'), B), \text{pf}_{\mathcal{V}'}(\text{mo}(S'), B)) \supseteq (\text{cl}(S'), \text{mo}(S')) \end{aligned} \tag{3}$$

Suppose that we are iteratively applying Step 1 of the Tracking Protocol, finding a sequence of isolated invariant sets where adjacent ones share an index pair, and we terminate with an isolated invariant set S and an index pair (P, E) . We can connect (P, E) with $(\text{cl}(S), \text{mo}(S))$ with techniques from the previous section. That is, if $(P, E) \neq (\text{cl}(S), \text{mo}(S))$, then we can find a filtration that connects them:

$$(P, E) \supseteq (P \cap \text{pf}_{\mathcal{V}}(\text{cl}(S), P), E \cap \text{pf}_{\mathcal{V}}(\text{mo}(S), P)) \subseteq (\text{pf}_{\mathcal{V}}(\text{cl}(S), P), \text{pf}_{\mathcal{V}}(\text{mo}(S), P)) \supseteq (\text{cl}(S), \text{mo}(S_1)) \quad (4)$$

We can then concatenate this filtration with the zigzag filtration in Equation 3. This effectively completes the Tracking Protocol: when continuation, represented as Step 1, is impossible, we can attempt to apply Step 2f and persistence to continue to track.

In Step 2f, we choose to take $S' = \text{Inv}_{\mathcal{V}'}(A)$. In practice, there may be many isolated invariant sets under \mathcal{V}' that are adjacent to S . However, our choice of S' is canonical.

► **Proposition 23.** *Let S' denote an isolated invariant set under \mathcal{V}' that is obtained from applying Step 2f of the Tracking Protocol to the isolated invariant set S under \mathcal{V} . If S'' is an isolated invariant set under \mathcal{V}' where $S \subseteq S''$, then $S' \subseteq S''$.*

Proof. By Proposition 4, set S'' is convex and \mathcal{V}' -compatible. Since A is the minimal convex and \mathcal{V}' -compatible set containing S we get that $S \subseteq A \subseteq S''$. By definition, $S' = \text{Inv}_{\mathcal{V}'}(A)$, so $S' \subseteq A \subseteq S''$. ◀

6 Conclusion

We conclude by briefly discussing some directions for future work. In Step 2g of the Tracking Protocol, there is a canonical choice of S' . But, as there is no common isolating set for S and S' , we cannot presently say anything about the persistence of the Conley index from S to S' . Is it possible to compute the Conley index persistence here in a controlled way? Can we meaningfully compute persistence for a different choice of invariant set? Investigation and experiments are likely needed to determine the most practical course of action in this case.

References

- 1 Madjid Allili, Tomasz Kaczynski, Claudia Landi, and Filippo Masoni. Acyclic partial matchings for multidimensional persistence: Algorithm and combinatorial interpretation. *J. Math. Imaging Vision*, 61:174–192, 2019. doi:10.1007/s10851-018-0843-8.
- 2 Bogdan Batko, Tomasz Kaczynski, Marian Mrozek, and Thomas Wanner. Linking combinatorial and classical dynamics: Conley index and Morse decompositions. *Found. Comput. Math.*, 20(5):967–1012, 2020.
- 3 Gunnar Carlsson and Vin de Silva. Zigzag persistence. *Found. Comput. Math.*, 10(4):367–405, August 2010. doi:10.1007/s10208-010-9066-0.
- 4 Charles Conley. Isolated invariant sets and the Morse index. In *CBMS Reg. Conf. Ser. Math.*, volume 38, 1978.
- 5 Tamal K. Dey, Marian Mrozek, and Ryan Slechta. Persistence of the Conley index in combinatorial dynamical systems. In *Proceedings of the 36th International Symposium on Computational Geometry*, pages 37:1–37:17, June 2020. doi:10.4230/LIPIcs.SocG.2020.37.
- 6 Tamal K. Dey, Marian Mrozek, and Ryan Slechta. Persistence of Conley-Morse graphs in combinatorial dynamical systems. *SIAM J. Appl. Dyn. Syst.*, 2022. To appear.
- 7 Tamal K. Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2022. URL: <https://www.cs.purdue.edu/homes/tamaldey/book/CTDAbook/CTDAbook.pdf>.

- 8 Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, January 2010.
- 9 Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, November 2002. doi:10.1007/s00454-002-2885-2.
- 10 Robin Forman. Combinatorial vector fields and dynamical systems. *Math. Z.*, 228:629–681, 1998. doi:10.1007/PL00004638.
- 11 Robin Forman. Morse theory for cell complexes. *Adv. Math.*, 134:90–145, 1998. doi:10.1006/aima.1997.1650.
- 12 David Gunther, Jan Reininghaus, Ingrid Hotz, and Hubert Wagner. Memory-efficient computation of persistent homology for 3d images using discrete Morse theory. In *2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 25–32, 2011. doi:10.1109/SIBGRAPI.2011.24.
- 13 Allen Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2002.
- 14 Tomasz Kaczynski, Marian Mrozek, and Thomas Wanner. Towards a formal tie between combinatorial and classical vector field dynamics. *J. Comput. Dyn.*, 3(1):17–50, 2016. doi:10.3934/jcd.2016002.
- 15 Henry King, Kevin Knudson, and Neža Mramor. Generating discrete Morse functions from point data. *Exp. Math.*, 14:435–444, 2005.
- 16 Kevin Knudson. *Morse Theory Smooth and Discrete*. World Scientific, 2015.
- 17 Kevin Knudson and Bei Wang. Discrete Stratified Morse Theory: A User’s Guide. In *34th International Symposium on Computational Geometry (SoCG 2018)*, volume 99, pages 54:1–54:14, 2018. doi:10.4230/LIPIcs.SoCG.2018.54.
- 18 Claudia Landi and Sara Scaramuccia. Relative-perfectness of discrete gradient vector fields and multi-parameter persistent homology. *J. Comb. Optim.*, 2021.
- 19 Michał Lipiński. *Morse-Conley-Forman theory for generalized combinatorial multivector fields on finite topological spaces*. PhD thesis, Jagiellonian University, 2021.
- 20 Michał Lipiński, Jacek Kubica, Marian Mrozek, and Thomas Wanner. Conley-Morse-Forman theory for generalized combinatorial multivector fields on finite topological spaces, 2020. arXiv:1911.12698.
- 21 Marian Mrozek. Conley–Morse–Forman theory for combinatorial multivector fields on Lefschetz complexes. *Found. Comput. Math.*, 17(6):1585–1633, December 2017. doi:10.1007/s10208-016-9330-z.
- 22 Marian Mrozek, Roman Srzednicki, Justin Thorpe, and Thomas Wanner. Combinatorial vs. classical dynamics: Recurrence. *Commun. Nonlinear Sci. Numer. Simul.*, 108:106226(1–30), 2022. doi:10.1016/j.cnsns.2021.106226.
- 23 Marian Mrozek and Thomas Wanner. Creating semiflows on simplicial complexes from combinatorial vector fields. *J. Differential Equations*, 304:375–434, 2021.
- 24 James Munkres. *Topology*. Featured Titles for Topology Series. Prentice Hall, Incorporated, 2000.
- 25 Vanessa Robins, Peter John Wood, and Adrian P. Sheppard. Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1646–1658, 2011. doi:10.1109/TPAMI.2011.95.

On the Discrete Fréchet Distance in a Graph

Anne Driemel ✉

Hausdorff Center for Mathematics, Universität Bonn, Germany

Ivor van der Hoog ✉

Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Lyngby, Denmark

Eva Rotenberg ✉ 

Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Lyngby, Denmark

Abstract

The Fréchet distance is a well-studied similarity measure between curves that is widely used throughout computer science. Motivated by applications where curves stem from paths and walks on an underlying graph (such as a road network), we define and study the Fréchet distance for paths and walks on graphs. When provided with a distance oracle of G with $O(1)$ query time, the classical quadratic-time dynamic program can compute the Fréchet distance between two walks P and Q in a graph G in $O(|P| \cdot |Q|)$ time. We show that there are situations where the graph structure helps with computing Fréchet distance: when the graph G is planar, we apply existing (approximate) distance oracles to compute a $(1 + \varepsilon)$ -approximation of the Fréchet distance between any shortest path P and any walk Q in $O(|G| \log |G| / \sqrt{\varepsilon} + |P| + \frac{|Q|}{\varepsilon})$ time. We generalise this result to near-shortest paths, i.e. κ -straight paths, as we show how to compute a $(1 + \varepsilon)$ -approximation between a κ -straight path P and any walk Q in $O(|G| \log |G| / \sqrt{\varepsilon} + |P| + \frac{\kappa|Q|}{\varepsilon})$ time. Our algorithmic results hold for both the strong and the weak discrete Fréchet distance over the shortest path metric in G .

Finally, we show that additional assumptions on the input, such as our assumption on path straightness, are indeed necessary to obtain truly subquadratic running time. We provide a conditional lower bound showing that the Fréchet distance, or even its 1.01-approximation, between arbitrary paths in a weighted planar graph cannot be computed in $O((|P| \cdot |Q|)^{1-\delta})$ time for any $\delta > 0$ unless the Orthogonal Vector Hypothesis fails. For walks, this lower bound holds even when G is planar, unit-weight and has $O(1)$ vertices.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Fréchet, graphs, planar, complexity analysis

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.36

Related Version *Full Version*: <https://arxiv.org/abs/2201.02121>

Funding Partially supported by Independent Research Fund Denmark grants 2020-2023 (9131-00044B) “Dynamic Network Analysis”.

Acknowledgements We thank David Goeckede and Petra Mutzel for useful discussions.

1 Introduction

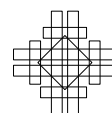
The Fréchet distance is a popular metric for measuring the similarity between (polygonal) curves. The Fréchet distance is often intuitively defined through the following metaphor: suppose that we have two curves that are traversed by a person and their dog. Over all possible traversals by both the person and the dog, what is the minimum length of their connecting leash? The Fréchet distance has many applications; in particular in the analysis and visualization of movement data [10, 14, 31, 44]. It is a versatile distance measure that can be used for a variety of objects, such as handwriting [38], coastlines [34], outlines of geometric shapes in geographic information systems [20], trajectories of moving objects,



© Anne Driemel, Ivor van der Hoog, and Eva Rotenberg;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 36; pp. 36:1–36:18
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



such as vehicles, animals or sports players [37, 39, 6, 14], air traffic [5] and also protein structures [28]. There are many variants of the Fréchet distance, some of which we also discuss further below. The two most-studied variants are the *continuous* and *discrete* Fréchet distance (based on whether the entities traverse a curve continuously or vertex-by-vertex).

Alt and Godau [2] were the first to study the Fréchet distance from a computational perspective. They studied how to compute the continuous Fréchet distance between two polygonal curves of n and m vertices each in $O(mn \log(n+m))$ time. Recently, this running time was improved by Buchin *et al.* [11] to $O(n^2 \sqrt{\log n} (\log \log n)^{3/2})$ on a real-valued pointer machine and $O(n^2 \log \log n)$ on a word RAM with word size $\Omega(\log n)$. Eiter and Maniá [23] showed how to compute the discrete Fréchet distance between two polygonal curves in $O(nm)$ time, which was later improved to $O(nm(\log \log nm)/\log nm)$ by Buchin *et al.* [11].

Conditional lower bounds for the Fréchet distance. The above (near-) quadratic upper bound algorithms are accompanied by a series of conditional lower bounds for computing the Fréchet distance or a constant factor approximation. All these results assume the Orthogonal Vector Hypothesis (OVH) or, by extension, the strong exponential time hypothesis (SETH) [42]. Bringmann [7] shows that there is no $O(n^{2-\delta})$ algorithm, for any $\delta > 0$, for computing the (discrete or continuous) Fréchet distance between two polygonal curves of n vertices each. The statement also holds for approximation algorithms with small constant approximation factor. Bringmann’s original proof uses self-intersecting curves in the plane. Later, Bringmann and Mulzer [9] showed the same conditional lower bound for intersecting curves in \mathbb{R}^1 . Bringmann [7] also showed the following conditional lower bound tailored to the unbalanced setting where the two input curves have different complexities: given two polygonal curves of n and m vertices each, there is no $O((nm)^{1-\delta})$ time algorithm for computing the Fréchet distance. Recently Buchin, Ophelders and Speckmann [13] showed that (assuming OVH) there can be no $O((nm)^{1-\delta})$ time algorithm that computes anything better than a 3-approximation of the Fréchet distance for pairwise disjoint planar curves in \mathbb{R}^2 and intersecting curves in \mathbb{R}^1 .

Avoiding lower bounds. These lower bounds can be circumvented whenever the input curves come from well-behaved classes of curves, such as c -packed curves [22, 8], ϕ -low density curves [22], and κ -straight curves [3, 4], and in special cases when the edges of the input curves are long [26]. Another way to avoid the quadratic complexity is to allow relatively large approximation factors. Bringmann and Mulzer [9] presented an α -approximation algorithm for the discrete Fréchet distance, that runs in time $O(n \log n + n^2/\alpha)$, for any α in $[1, n]$. This was recently improved by Chan and Rahmati [16] to $O(n \log n + n^2/\alpha^2)$ for any α in $[1, n/\log n]$. For the continuous Fréchet distance a weaker result was presented by Colombe and Fox [19]. They show an $O(\alpha)$ -approximation algorithm for any α in $[\sqrt{n}, n]$ that runs in time $O((n^3/\alpha^2) \log n)$. For general polygonal curves, without further input assumptions, the best-known approximation factors with near-linear running times are still quite high, $\alpha \approx n$ for the continuous Fréchet distance and $\alpha \approx \sqrt{n}$ for the discrete case.

Fréchet distance variants. Variants of the Fréchet distance include those that model partial similarity by allowing straight-line shortcuts along a curve [21], or by maximizing the portions of the curves that are matched to each other within a fixed distance [12]. Other variants constrain the class of mappings by applying speed constraints [33] or topological constraints [15], or model the distance metric to the geodesics inside a simple polygon [27]. Even other variants extend the class of mappings, such as the weak Fréchet distance, which



■ **Figure 1** (a) A road network can be represented as a graph G . (b) Edges in G can be weighted, e.g. depending on whether traffic flows fast (grey) or slow (black). Under the shortest path metric, the Fréchet distance between blue and green may be smaller than the distance between red and black; even though under the Euclidean metric, the red-black Fréchet distance is smaller.

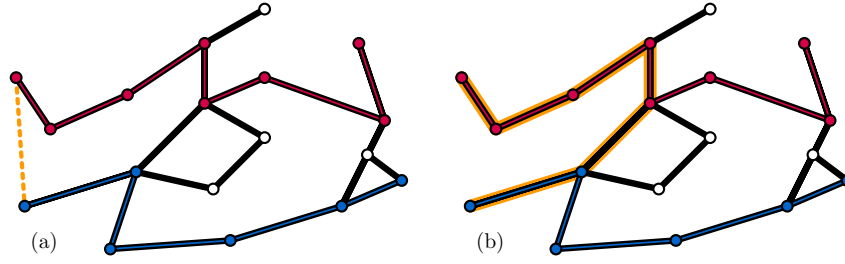
was already studied by Alt and Godau [2]. Strikingly, the Fréchet distance has not been studied in the context of graphs. Edge-weighted graphs with their shortest-path metric are commonly used to model discrete metric spaces [35], and the Fréchet distance can be derived from the underlying distance metric (Figure 2). In this paper, we intend to initiate a study of the computational complexity of the discrete Fréchet distance between paths in a planar graph, where distances between nodes are measured by their shortest path metric in this graph. This is a natural model when, for example, measuring the similarity of two trajectories in the same street network (Figure 1).

Contribution and organisation. This is the first paper that considers computing the Fréchet distance in the graph domain.¹ Section 2 contains the preliminaries where we present an overview of distance oracles and the problem statement. Section 3 serves as an introduction to our setting and techniques. We assume that P is a κ -straight path and that Q is a walk in a planar weighted graph G . We use an exact distance oracle with $O(\log^{2+o(1)} |G|)$ query time to compute a $(\kappa + 1)$ -approximation of $D_{\mathcal{F}}(P, Q)$. This is the first nontrivial algorithm for computing the (approximate) Fréchet distance in a planar graph. In Section 4 we extend our results. We use a $(1 + \alpha)$ -stretch distance oracle to compute a $(1 + \varepsilon)$ -approximation of $D_{\mathcal{F}}(P, Q)$. The full version contains the analogous result for the weak Fréchet distance. Finally, we show in Section 5 a conditional lower bound for computing the Fréchet distance. Specifically, assuming the Orthogonal Vector Hypothesis (OVH), we show that if G is an integer-weighted planar graph, P and Q are paths in G and $m = n^\gamma$ for some constant $\gamma > 0$, then for every $\delta > 0$ there can be no algorithm that computes $D_{\mathcal{F}}(P, Q)$ (or a 1.01-approximation) in $O((nm)^{1-\delta})$ time unless OVH fails. In the full version we consider walks P and Q in a planar unit-weight graph with a constant number of vertices.

2 Preliminaries

Let $G = (V, E)$ be a planar undirected weighted graph with N vertices, where every edge e_i has some corresponding integer weight ω_i and all weights can be expressed in a word of $\Theta(\log N)$ bits. For any two vertices $v_1, v_2 \in V$ their distance, denoted by $d(v_1, v_2)$, is the

¹ Similar ideas were used in the master's thesis of David Goeckede [24]. In particular, the approach we use in Section 3 and a lower bound construction for walks was used there.



■ **Figure 2** The Fréchet distance may be derived from the Euclidean or the shortest path metric.

length of the shortest path from v_1 to v_2 in G . A walk in G is any sequence of vertices where every subsequent pair of vertices is connected by an edge in E . A path in G is a walk where no vertex appears twice in the sequence. Let P be any walk in G , represented by an ordered set of vertices $P = (p_1, p_2, \dots, p_n)$. We denote by $|P| = n$ the number of vertices in P and by $[n]$ the set $(1, 2, \dots, n)$. We denote the walk $Q = (q_1, q_2, \dots, q_m)$, $|Q|$ and $[m]$ analogously.

Discrete Fréchet distance. Given two walks P and Q in G , we denote by $[n] \times [m] \subset \mathbb{N} \times \mathbb{N}$ the integer lattice of n by m integers. We say that an ordered sequence F of points in $[n] \times [m]$ is a *discrete walk* if for every consecutive pair $(i, j), (k, l) \in F$, we have $k \in \{i - 1, i, i + 1\}$ and $l \in \{j - 1, j, j + 1\}$. It is furthermore *xy-monotone* when we restrict to $k \in \{i, i + 1\}$ and $l \in \{j, j + 1\}$. Let F be a discrete walk from $(1, 1)$ to (n, m) . The *cost* of F is the maximum over $(i, j) \in F$ of $d(p_i, q_j)$. The (weak) discrete Fréchet distance is the minimum over all (not necessarily *xy-monotone*) walks F from $(1, 1)$ to (n, m) of its associated cost:

$$D_{\mathcal{F}}(P, Q) := \min_F \text{cost}(F) = \min_F \max_{(i,j) \in F} d(p_i, q_j).$$

The discrete free-space matrix. In this paper we show an algorithm for computing the discrete Fréchet distance between two walks P and Q in a graph G . To this end, we use what we will call a free-space matrix which can be seen as a discrete free-space diagram. Given P, Q and some real value ρ , we construct a $|P| \times |Q|$ matrix M which we call the free-space matrix M_ρ . The i 'th column of M_ρ corresponds to the vertex $p_i \in P$ and the j 'th row corresponds $q_j \in Q$. We assign to each matrix cell $M_\rho[i, j]$ the integer -1 if $d(p_i, q_j) \leq \rho$, and a 0 if $d(p_i, q_j) > \rho$. From our above definition of the discrete Fréchet distance, we immediately conclude the following:

► **Lemma 1.** *The Fréchet distance between P and Q is at most ρ , if and only if there exists a discrete (*xy-monotone*) walk F from $(1, 1)$ to (n, m) such that $\forall (i, j) \in F, M_\rho[i, j] = -1$.*

Orthogonal Vectors Hypothesis. The Orthogonal Vectors problem can be stated as follows. Given are a set A and B of d -dimensional Boolean vectors with $|A| = n$ and $|B| = m$. The goal is to identify whether there exist two vectors $a = (a_1, a_2, \dots, a_d)$ and $b = (b_1, b_2, \dots, b_d)$ with $a \in A$ and $b \in B$, such that a and b are orthogonal (i.e. $\sum_{i=1}^d a_i \cdot b_i = 0$). In this paper, we use the following variant of the Orthogonal Vectors hypothesis. It is implied by SETH, see Abboud and Williams [1, Section 3], and it is equivalent to the standard variant of OVH defined by Williams [42], see Bringmann [7].

► **Definition 2.** *The Orthogonal Vectors Hypothesis states that for every $\delta > 0$ and $1 > \gamma > 0$, there exists an $\omega > 0$ and such that the Orthogonal Vectors problem for d -dimensional vectors with $d = \omega \log n$ and $m = n^\gamma$, cannot be solved in $O((nm)^{1-\delta})$ time.*

Distance oracles. A distance oracle is a compact data structure that facilitates fast exact or approximate distance queries between vertices in a graph. A distance oracle has *stretch* S if it never underestimates the distance, and it at most overestimates by a factor S , i.e. $d(a, b) \leq d_{\text{estim.}}(a, b) \leq S \cdot d(a, b)$. For general graphs [36, 41, 43], the best possible stretch in sub-quadratic space is 3, but for planar graphs on N vertices, Thorup [40] shows that it is possible to compute $(1 + \varepsilon)$ -stretch distance oracles in the near-linear $O(N/\varepsilon \log N)$ time and space, and with a query-time of $O(1/\varepsilon)$. The study of distance oracles for planar graphs is an active research area [17, 18, 25, 29, 30, 32, 40]. For $(1 + \varepsilon)$ -stretch oracles, Gu and Xu [25] show that it is possible to achieve constant query-time *independently of* ε at the cost of an increased construction time and space of $O(N(\log N)^4/\varepsilon + 2^{O(1/\varepsilon)})$. Even for exact distances, Charalampopoulos et al. [17] give an $O(N^{1+o(1)})$ -space and $O(N^{o(1)})$ -query time data structure. Long and Pettie [32] improve these exact queries to polylogarithmic $O((\log(N))^{2+o(1)})$ time while maintaining the $O(N^{1+o(1)})$ -space bound.

In the following sections we use the exact distance oracle by Long and Pettie [32] and the $(1 + \varepsilon)$ -stretch oracle by Thorup [40]. Any distance oracle that improves the efficiency of these data structures, or any extension of them to larger classes of graphs, immediately leads to improving or extending our results correspondingly.

From distance oracles to an upper bound. Given a distance oracle with $T(G)$ query time it is straightforward to find an $O(nm \cdot T(G))$ time algorithm for computing $D_{\mathcal{F}}(P, Q)$ between two walks P and Q in G that “matches” the conditional $\Omega(nm^{1-\delta})$ lower bound. Indeed, for any pair $(p, q) \in P \times Q$ we can query their pairwise distance in G . Given such a weighted graph, we want to find an xy -monotone path from $(1, 1)$ to (n, m) with minimal cost (which can be done with an $O(nm \cdot T(G))$ dynamic program as by Eiter and Manila [23]).

κ -straight paths. Alt, Knauer and Wenk [3] define κ -straight paths as a generalisation of shortest paths. A path P is κ -straight if for any two points $s, t \in P$, the length of the subpath $P[s, t]$ from s to t is at most $\kappa \cdot d(s, t)$. Shortest paths are 1-straight. When we replace the term “points” by “vertices”, this definition immediately transfers to our graph setting.

3 A $(\kappa + 1)$ -approximation for the discrete Fréchet distance

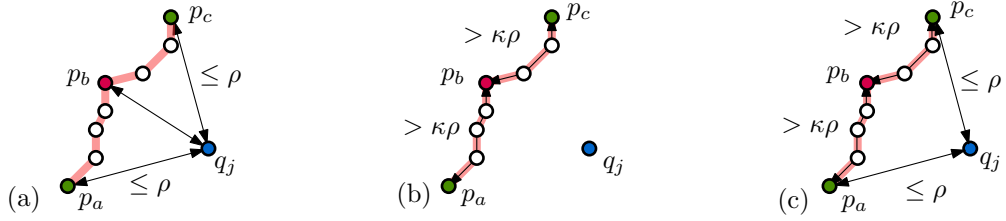
Let $G = (V, E)$ be a planar weighted graph with N vertices and integer weights. We use the structure by Long and Pettie [32] to preprocess G , such that given two walks $P = (p_1, \dots, p_n)$ and $Q = (q_1, \dots, q_m)$, where P is a κ -straight path we can compute a $(\kappa + 1)$ -approximation of $D_{\mathcal{F}}(P, Q)$. In the following section we extend this approach to an algorithmic result for computing a $(1 + \varepsilon)$ -approximation. Recall that the decision variant of the Fréchet distance may be answered with the help of a free-space matrix M_{ρ} . Here, we extend its definition:

► **Definition 3.** We denote by M_{ρ}^{κ} the κ -straight free-space matrix, which is a matrix with dimensions $n \times m$. We define the matrix $M_{\rho}^{\kappa}[i, j]$ as follows:

- $M_{\rho}^{\kappa}[i, j] = -1$ if the distance $d(p_i, q_j) \leq \rho$,
- $M_{\rho}^{\kappa}[i, j] = 1$ if the distance $d(p_i, q_j) > (\kappa + 1)\rho$, or
- $M_{\rho}^{\kappa}[i, j] = 0$ otherwise.

Every cell $M_{\rho}^{\kappa}[i, j]$ has a corresponding point (i, j) in the integer lattice $[n] \times [m]$. The discrete Fréchet distance is at most ρ , iff there exists a discrete walk F through $[n] \times [m]$ where for every pair $(i, j) \in F$, $M_{\rho}^{\kappa}[i, j] = -1$. Explicitly constructing M_{ρ}^{κ} takes at least $\Omega(nm)$ time. However, we show that we can use the distance oracle to implicitly traverse M_{ρ}^{κ} to find the existence of such a discrete walk. To this end, we first show the following:

36:6 On the Discrete Fréchet Distance in a Graph



■ **Figure 3** (a) Three vertices $p_a, p_b, p_c \in P$ and a vertex $q_j \in Q$ such that $M_\rho^\kappa[a, j] = M_\rho^\kappa[c, j] = -1$ and $M_\rho^\kappa[b, j] = 1$. (b) We show that the distance between p_a and p_b must be more than $\kappa\rho$. (c) However, this implies that P is not κ -straight, as there is a shortcut from p_a to p_c through q_j .

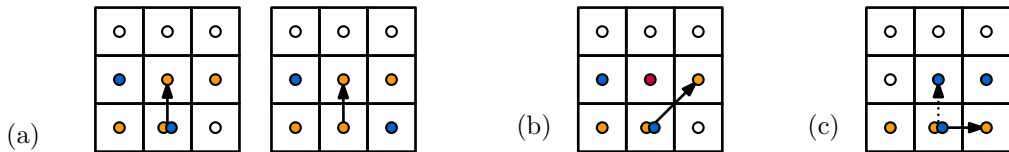
► **Lemma 4.** Let P be a κ -straight path and Q a walk in G , ρ be some fixed value and $j \leq m$ some integer. For any two integers a, c such that $M_\rho^\kappa[a, j] = -1$ and $M_\rho^\kappa[c, j] = -1$, there cannot be an integer $b \in [a, c]$ for which $M_\rho^\kappa[b, j] = 1$.

Proof. Suppose for the sake of contradiction that there are three integers a, b, c with $b \in [a, c]$, $M_\rho^\kappa[a, j] = -1$ and $M_\rho^\kappa[c, j] = -1$ and $M_\rho^\kappa[b, j] = 1$. It cannot be that $b = a$ or $b = c$, so there are three vertices $p_a, p_b, p_c \in P$ with $d(p_a, q_j) \leq \rho$, $d(p_c, q_j) \leq \rho$ and $d(p_b, q_j) > (\kappa + 1)\rho$ (Figure 3). Moreover, p_b lies on the κ -straight subpath $P[p_a, p_c]$. It follows that the length of the subtrajectory $P[p_a, p_b]$ is more than $\kappa\rho$ (otherwise, the distance between p_b and q_j is at most $(\kappa + 1)\rho$ by the path through p_a to q_j). We can apply a symmetric argument to $P[p_b, p_c]$. Thus, the length of $P[p_a, p_c]$ is more than $2\kappa\rho$. At the same time, there exists a path in G from p_a to p_b through q_j of length at most 2ρ . This contradicts that P is κ -straight. ◀

A consequence of the above lemma is the following: let (i, j) be a lattice point for which $M_\rho^\kappa[i, j] = -1$. For the nearest lattice point (l, j) left of (i, j) for which $M_\rho^\kappa[l, j] = 1$, there can be no lattice point left of (l, j) for which the matrix evaluates to -1 . A symmetrical statement holds for the nearest such point right of (i, j) . This leads to the following algorithm to conclude if $D_{\mathcal{F}}(P, Q) \leq (\kappa + 1)\rho$ or $D_{\mathcal{F}}(P, Q) > \rho$, where we construct a discrete walk F' :

We compute the distance oracle in $O(N^{1+o(1)})$ time. If $M_\rho^\kappa[1, 1] > -1$ then our algorithm terminates and concludes that $D_{\mathcal{F}}(P, Q) > \rho$. We iteratively perform the following procedure, to construct a path F' . Let (i, j) be the latest point added to F' , then:

1. If $(i, j) = (n, m)$ the algorithm terminates and concludes that $D_{\mathcal{F}}(P, Q) \leq (\kappa + 1)\rho$.
2. If $(j + 1) > m$, go to the last step.
3. Otherwise, we use two distance queries to check $M_\rho^\kappa[i, j + 1]$ and $M_\rho^\kappa[i + 1, j + 1]$:
 - (i) If $M_\rho^\kappa[i, j + 1] = -1$, add $(i, j + 1)$ to F' .
 - (ii) Else if $M_\rho^\kappa[i + 1, j + 1] = -1$, add $(i + 1, j + 1)$ to F' .
4. Otherwise, we use a distance query to check if $M_\rho^\kappa[i + 1, j]$:
 - (i) If $(i + 1) > n$ or $M_\rho^\kappa[i + 1, j] = 1$, we terminate the procedure and conclude that $D_{\mathcal{F}}(P, Q) > \rho$.
 - (ii) Otherwise, we add $(i + 1, j)$ to F' .



■ **Figure 4** Lattice points to prove Lemma 5. Blue $\in F$. Orange $\in F'$ and Red $\notin F$.

► **Lemma 5.** *Let P be κ -straight in G , Q be any walk and $D_{\mathcal{F}}(P, Q) < \rho$. Denote by F an xy -monotone path over the lattice $[n] \times [m]$ such that for all $(i, j) \in F$, $M[i, j] = -1$. All lattice points in our constructed path F' are either in F or lie to the left of a point of F .*

Proof. Consider for the sake of contradiction the first iteration where the algorithm would add a lattice point (c, d) right of a point in F . Let $(a, b) \in F'$ be the point preceding (c, d) . We make a case distinction based on whether (c, d) was added through step 3(i), 3(ii) or 4(ii). The three cases are illustrated by Figure 4, (a) (b) and (c) respectively.

First suppose that $(c, d) = (a, b + 1)$. Since (c, d) is the first point right of F , it must be that F contains either (a, b) or a point right of (a, b) . Moreover (since (c, d) is right of F), F also contains a point left of $(a, b + 1)$. This implies that F is not xy -monotone, contradiction.

Now suppose that $(c, d) = (a + 1, b + 1)$. Because we reached step 3(ii), we know that $M_{\rho}^{\kappa}[a, b + 1] > -1$ and thus $(a, b + 1) \notin F$. However, since (c, d) is the first point right of F , F either contains (a, b) or a point right of (a, b) , and a point strictly left of $(a, b + 1)$. This implies that F is not xy -monotone which is a contradiction.

Finally, suppose that $(c, d) = (a + 1, b)$. Since (c, d) is the first point right of F , it must be that $(a, b) \in F$. However, consider now the successor of (a, b) in F . Since F is xy -monotone, this successor is either $(a, b + 1)$ or $(a + 1, b + 1)$, as it cannot be $(a + 1, b) = (c, d)$. However, this implies that either $M_{\rho}^{\kappa}[a, b + 1] = -1$ or $M_{\rho}^{\kappa}[a + 1, b + 1] = -1$, which contradicts the assumption that we have reached step 4 of the algorithm. ◀

With these two observations, we are ready to prove our main theorem:

► **Theorem 6.** *We can preprocess a planar graph G with N vertices in $O(N^{1+o(1)})$ time and space s.t. for any κ -straight path $P = (p_1, \dots, p_n)$, walk $Q = (q_1, \dots, q_m)$ and $\rho \in \mathbb{R}$, we can conclude either $D_{\mathcal{F}}(P, Q) > \rho$ or $D_{\mathcal{F}}(P, Q) \leq (\kappa + 1)\rho$ in $O((n + m) \log^{2+o(1)} N)$ time.*

Proof. We first preprocess G to construct a distance oracle using $O(N^{1+o(1)})$ time and space. Given ρ , our algorithm spends at most $n + m$ iterations before it either reaches (n, m) or step 4(i) and terminates. At each iteration we perform at most three distance queries. We prove that if $D_{\mathcal{F}}(P, Q) \leq \rho$, we always conclude that $D_{\mathcal{F}}(P, Q) \leq (\kappa + 1)\rho$. Indeed, suppose that $D_{\mathcal{F}}(P, Q) \leq \rho$ then there exists a discrete walk F such that for every $(i, j) \in F$, $M_{\rho}^{\kappa}[i, j] = -1$ and F is xy -monotone. Per construction, the path F' is xy -monotone and for all $(i, j) \in F'$, $M[i, j] < 1$. What remains to show is that F' is from $(1, 1)$ to (n, m) . Suppose for the sake of contradiction that F' does not reach (n, m) and let (i, j) be the last element added to F' before the algorithm terminated in step 4. Since we reached step 4 it must be that:

$$M_{\rho}^{\kappa}[i, j + 1] > -1 \text{ and } M_{\rho}^{\kappa}[i + 1, j + 1] > -1 \quad (\text{or } (j + 1 \leq m)).$$

Let $\ell \leq i$ be the lowest integer such that $M_{\rho}^{\kappa}[\ell, j] = -1$. Such an ℓ must always exist, since we only enter the j 'th row through a point (k, j) for which $M_{\rho}^{\kappa}[k, j] = -1$ (step 3(i) or 3(ii)). Since we arrived in step 4(i), it must be that either $M_{\rho}^{\kappa}[i + 1, j] = 1$ or $(i + 1) > n$. However, this implies that $(i, j) \in F$ (indeed, by Lemma 5 there exists a point equal to or to the right of (i, j) in F). However, given Lemma 4 and (ℓ, i) , there is no a point in F right of (i, j) . Because if F is xy -monotone, the successor of $(i, j) \in F$ is either $(i + 1, j + 1)$, $(i + 1, j)$ or $(i, j + 1)$. Since we terminated, none of these elements can be in F , contradiction. ◀

The following corollary is a direct result of the assumption that edge weights each fit in a constant number of words (thus, the range of values for $D_{\mathcal{F}}(P, Q)$ is polynomial in N).

► **Corollary 7.** *We can preprocess a planar graph G with N vertices in $O(N^{1+o(1)})$ time such that: for any κ -straight path $P = (p_1, \dots, p_n)$ and walk $Q = (q_1, \dots, q_m)$, we can compute a $(\kappa + 1)$ -approximation of $\mathcal{D}(G)(P, Q)$ in $O((n + m) \log^{3+o(1)} N)$ time.*

4 A $(1 + \varepsilon)$ -approximation for Fréchet distance

We present a more involved approach to compute a $(1 + \varepsilon)$ approximation of $D_{\mathcal{F}}(P, Q)$. Specifically, we choose $(1 + \varepsilon) = (1 + \alpha)(1 + \alpha + \beta)$ for some α and β . We show for any ρ how to correctly conclude either $D_{\mathcal{F}}(P, Q) \leq (1 + \alpha)(1 + \alpha + \beta)\rho$ or $D_{\mathcal{F}}(P, Q) > \rho$.

To obtain this result, we use two data structures. A Voronoi diagram of P in G marks every vertex v in G with the closest vertex $p \in P$ (and the exact distance $d(v, p)$). For completeness, we prove in the full version the following (folklore) result:

► **Theorem 8.** *For any planar weighted graph $G = (V, E)$ and any vertex set $P \subseteq V$, it is possible to construct the Voronoi diagram of P in G in $O(|V| \log |V|)$ time.*

Additionally, we use the $(1 + \alpha)$ -stretch distance oracle $\mathcal{D}(G)$ by Thorup [40]. We differentiate between the distance $d(p_i, q_j)$ and what we call the *perceived* distance between p_i and q_j . For any two vertices p_i, q_j we denote by $d_o(p_i, q_j)$ their *perceived* distance (the result of the distance query of $\mathcal{D}(G)$). Per definition $d(p_i, q_j) \leq d_o(p_i, q_j) \leq (1 + \alpha) \cdot d(p_i, q_j)$.

► **Definition 9.** *For a given value $\rho \in \mathbb{R}$ we denote by M_{ρ}^{β} the approximate free-space matrix, which is a matrix with dimensions $n \times m$ where:*

- $M_{\rho}^{\beta}[i, j] = -1$ if the perceived distance $d_o(p_i, q_j) \leq (1 + \alpha)\rho$,
- $M_{\rho}^{\beta}[i, j] = 1$ if the perceived distance $d_o(p_i, q_j) > (1 + \alpha)(1 + \alpha + \beta)\rho$, or
- $M_{\rho}^{\beta}[i, j] = 0$ otherwise.

β -compression. Given a κ -straight path P and real values (ρ, β) we define the β -compression P^{β} as an ordered set that is obtained in three steps (Figure 5):

- The first step is a greedy iterative process where:
 - we remove (consecutive) p_x where the length of $P[p_1, p_x]$ is fewer than $\beta\rho$.
 - the first such vertex p_i that does not meet this criterion is added to P^{β} . Then, we remove (consecutive) p_x where the length of $P[p_i, p_x]$ is fewer than $\beta\rho$. and so forth.
- In the second step we add for every vertex in P^{β} its preceding vertex in P .
- In the third step we add p_n .

The result of this procedure is that we have an ordered set P^{β} with $n' \leq n$ vertices. We create a map $\pi : [n'] \leftrightarrow [n]$ that maps every vertex in P^{β} to its corresponding vertex in P (i.e. the k 'th element of P^{β} is denoted by $p_{\pi(k)} \in P$) and we observe:

- $\pi(1) = 1$ and $\pi(n') = n$,
- for all i , the length of $P[p_{\pi(i)}, p_{\pi(i+3)}]$ is greater than $\beta\rho$ and
- for all $x \in [\pi(i), \pi(i+1)]$, the exact distance $d(p_{\pi(i)}, p_x) < \beta\rho$ and $d(p_{\pi(i+1)}, p_x) < \beta\rho$.

We denote $P^{\beta} = (p_{\pi(1)}, p_{\pi(2)}, \dots, p_{\pi(n')})$. The global approach is to approximate the Fréchet distance between P^{β} and Q instead. We first note the following three properties of P^{β} :

► **Lemma 10.** *For every two integers i and j , if $M_{\rho}^{\beta}[\pi(i), j] = -1$, then for all integers $x \in (\pi(i-1), \pi(i+1))$ it must be that $M_{\rho}^{\beta}[i, j] \leq 1$.*

Proof. Either $p_{\pi(i-1)}$ and $p_{\pi(i)}$ are consecutive in P (thus, the set $(\pi(i-1), \pi(i))$ is empty) or per construction the length of $P[p_{\pi(i-1)}, p_{\pi(i)}]$ is less than $\beta\rho$.

Thus, if the perceived distance $d_o(p_{\pi(i)}, q_j) \leq (1 + \alpha)\rho$, then for all points p_x with $x \in (\pi(i-1), \pi(i))$, the exact distance $d(p_x, q_j) \leq (1 + \alpha + \beta)\rho$ by traversing through $p_{\pi(i)}$. Thus, the perceived distance $d_o(p_x, q_j) \leq (1 + \alpha)(1 + \alpha + \beta)\rho$. A symmetrical argument holds for all $x \in (\pi(i), \pi(i+1))$. ◀

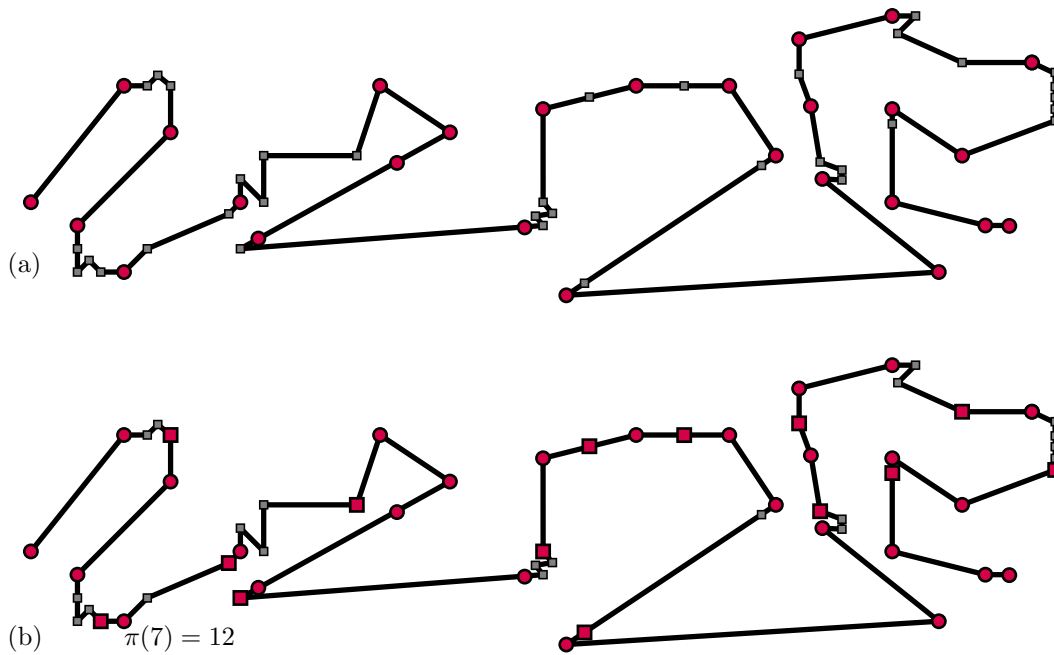


Figure 5 A planar path where the edge weights correspond to their length. (a) We greedily add vertices to P^β such that for all vertices $p_x \in P$ with preceding vertex $p_i \in P^\beta$ the length of $P[p_i, p_x]$ is at most $\beta\rho$. (b) For every vertex in P^β , we subsequently add its preceding vertex in P to P^β .

► **Lemma 11.** For all i and j , if there exists an integer $x \in (\pi(i), \pi(i+1))$ such that $M_\rho^\beta[x, j] = -1$, then $M_\rho^\beta[\pi(i), j] \leq 1$ and $M_\rho^\beta[\pi(i+1), j] \leq 1$.

Proof. As in Lemma 10, $d(p_x, p_{\pi(i)}) \leq \beta\rho$ and $d(p_x, p_{\pi(i+1)}) \leq \beta\rho$ implies the lemma. ◀

► **Lemma 12.** For any j , let i be an integer such that there exists an $x \in [\pi(i), \pi(i+1)]$ with $M_\rho^\beta[x, j] = -1$. Denote $a = i - \lceil \frac{9\kappa}{\beta} \rceil$ and $b = i + \lceil \frac{9\kappa}{\beta} \rceil$. There can be no integer $y \notin [\pi(a), \pi(b)]$ such that $M_\rho^\beta[y, j] = -1$.

Proof. For all i , the length of $P[p_{\pi(i)}, p_{\pi(i+3)}]$ is greater than $\beta\rho$. It follows that the length of the subpath $P[p_{\pi(a)}, p_x]$ is more than: $\sum_{t=1}^{\lceil \frac{3\kappa}{\beta} \rceil} \beta\rho = \frac{3\kappa}{\beta} \beta\rho = 3\kappa\rho$ (Figure 6). Suppose for the sake of contradiction that there exists an integer $y < \pi(a)$ such that $d_o(p_y, p_j) \leq (1+\alpha)\rho$. Then the exact distance $d(p_y, p_x)$ is at most $2(1+\alpha)\rho$ through traversing from p_y to p_j to p_x .

However, the subpath $P[p_y, p_x]$ is longer than $P[p_{\pi(a)}, p_x]$ and thus longer than $3\kappa\rho$. For $\alpha < 0.5$, this contradicts the assumption that P is κ -straight.

A symmetrical argument holds for $y > \pi(b)$. ◀

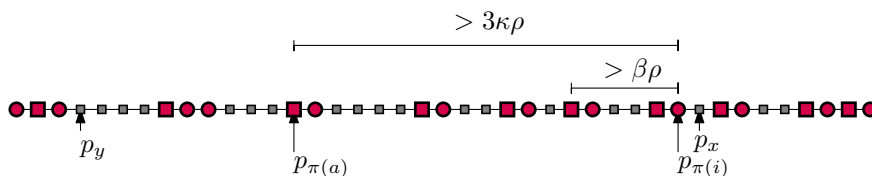


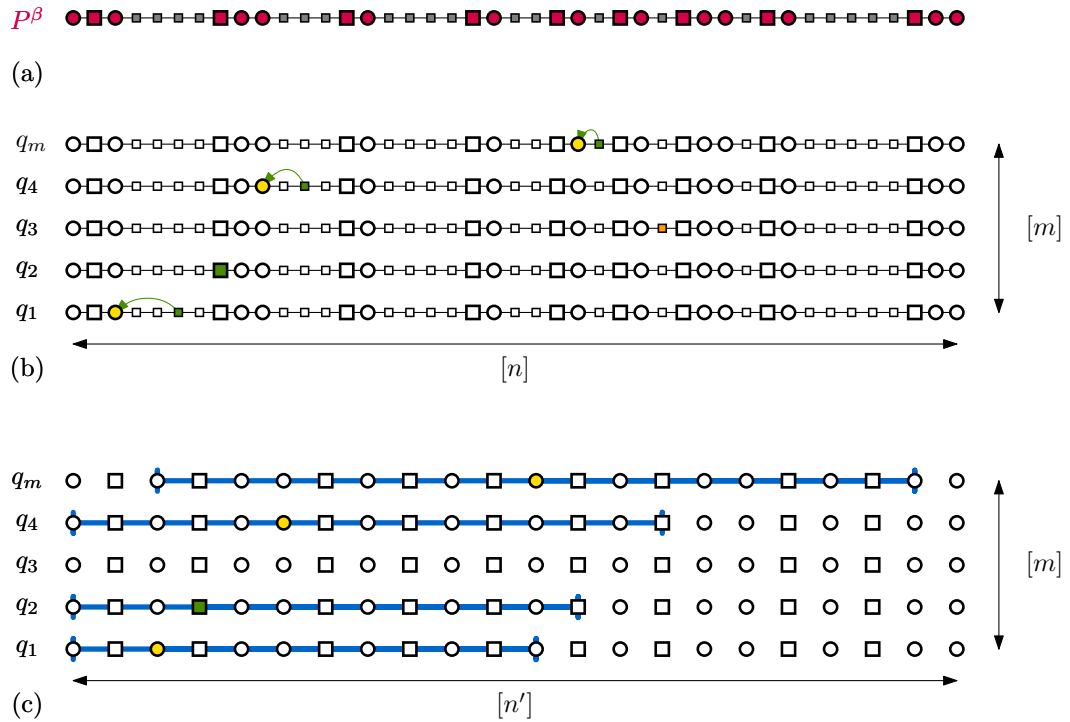
Figure 6 A schematic representation of P^β . For any i as in Lemma 12, we consider an integer $a = i - \lceil \frac{9\kappa}{\beta} \rceil$ and some p_y preceding $p_{\pi(a)}$.

36:10 On the Discrete Fréchet Distance in a Graph

Defining β -windows. Now, we use two lattices: $[n] \times [m]$ and the smaller lattice $[n'] \times [m]$. Points on the first lattice will be denoted by (x, j) and (y, j) . Points on the second lattice will be denoted by (i, j) or (a, j) or (b, j) . Intuitively, Lemma 12 shows for every integer j a “horizontal window” in $[n'] \times [m]$ (of width $O(\frac{\kappa}{\beta})$) that bounds the subpath of P of vertices that *may* have perceived distance fewer than $(1 + \alpha)\rho$ to the vertex $q_j \in Q$. We formalise this intuition by defining β -windows (denoted by W_1, W_2, \dots, W_m , see Figure 7):

- Let for an index j , p_x be any vertex in P with minimal distance to q_j in the graph G .
- Let i be the integer such that $p_{\pi(i)}$ is the point in P^β that precedes p_x .
- We distinguish two cases:
 1. If the exact distance $d(p_x, q_j) > \rho$ then: W_j is empty.
 2. Otherwise: $W_j = [i - \lceil \frac{9\kappa}{\beta} \rceil, i + \lceil \frac{9\kappa}{\beta} \rceil] \times \{j\} \subset [n'] \times [m]$.

The high-level approach. We first construct the Voronoi diagram of P in G in $O(N \log N)$ time. For every $q_j \in Q$, we obtain from the diagram the vertex $p_x \in P$ that is closest to q_j and the *exact* distance $d(p_x, q_j)$ in $O(1)$ time. With q_j , we construct W_j in $O(\frac{\kappa}{\beta})$ time. For every point $(a, j) \in W_j$ we compute $d(p_{\pi(a)}, j)$ in $O(\frac{1}{\alpha})$ time. Any lattice walk that realises a distance $D_{\mathcal{F}}(P, Q) \leq (1 + \alpha)(1 + \alpha + \beta)\rho$ must be contained in the grid: $A = \cup_j W_j$ which has $O(m \cdot \frac{\kappa}{\beta})$ complexity. We compute a minimal cost path in time linear in the size of A .



■ **Figure 7** (a) a schematic representation of a path P with P^β in red. (b) For every $j \in [m]$, we observe the closest point p_x . If $d(p_x, q_j) \leq \rho$ we color it green. Otherwise, we color it orange. In addition, if $p_x \notin P^\beta$ we color its predecessor in P^β yellow. (c) For every yellow or green vertex in $[n'] \times [m]$, we create a horizontal window in blue. We show the window for $\kappa = \beta = 1$.

► **Theorem 13.** Let G be a planar graph with N vertices, $P = (p_1, \dots, p_n)$ a κ -straight path and $Q = (q_1, \dots, q_m)$ be any walk in G . Given a value $\rho \in \mathbb{R}$ and some β and $\alpha \leq 0.5$, we correctly conclude either $D_{\mathcal{F}}(P, Q) > \rho$ or $D_{\mathcal{F}}(P, Q) \leq (1 + \alpha)(1 + \alpha + \beta)\rho$ in $O(N \log N / \alpha + n + \frac{\kappa}{\alpha\beta}m)$ time using $O(N \log N / \alpha)$ space.

Proof. We construct the approximate distance oracle $\mathcal{D}(G)$ using $O(N \log N/\alpha)$ time and space. Given P and Q , we construct the β -compressed path P^β in $O(n)$ time. We supply every point in $P \setminus P^\beta$ with a pointer to the point in P^β that precedes it. We construct the Voronoi diagram of P in the graph G in $O(N \log N)$ time. Given P^β , we construct for every integer $j \in [m]$ the window W_j in $O(\frac{\kappa}{\beta})$ time. Specifically, for any point q_j we obtain the point p_x that is closest to q_j . If $d(p_x, q_j) \leq \rho$ then we obtain the point $p_{\pi(i)}$ in P^β that precedes p_x in constant time through the pre-stored pointer and we set: $W_j = [i - \lceil \frac{9\kappa}{\beta} \rceil, i + \lceil \frac{9\kappa}{\beta} \rceil] \times \{j\}$.

The union of windows ($A = \cup_j W_j$) is a grid in $[n'] \times [m]$ of at most $O(m \cdot \frac{\kappa}{\beta})$ lattice points. For each $(a, j) \in A$ we query $\mathcal{D}(G)$ in $O(\frac{1}{\alpha})$ time to determine the value $M_\rho^\beta[\pi(a), j]$ in $O(m \frac{\kappa}{\alpha\beta})$ total time. Given this grid, we construct a directed grid graph where there is:

- a vertical edge from (a, j) to $(a, j + 1)$ if $M_\rho^\beta[\pi(a), j] < 1$ and $M_\rho^\beta[\pi(a), j + 1] < 1$,
 - a horizontal edge from (a, j) to $(a + 1, j)$ if $M_\rho^\beta[\pi(a), j] < 1$ and $M_\rho^\beta[\pi(a + 1), j] = -1$,
 - diagonal edge from (a, j) to $(a + 1, j + 1)$ if $M_\rho^\beta[\pi(a), j] < 1$ and $M_\rho^\beta[\pi(a + 1), j + 1] = -1$.
- We can determine if there exists a path in A from $(1, 1)$ to (n', m) in $O(\frac{m\kappa}{\beta})$ time.

If such a path F^* exists. we claim that $D_{\mathcal{F}}(P, Q) \leq (1 + \alpha)(1 + \alpha + \beta)\rho$. Indeed, we transform F^* into a path over $[n] \times [m]$ as follows: for all $(a, j) \in F^*$ we add $(\pi(a), j)$. Note that per construction of the grid graph, for all points in F^* it must be that $M_\rho^\beta[\pi(a), j] < 1$ and thus $d_o(\pi(a), j) \leq (1 + \alpha)(1 + \alpha + \beta)\rho$. For every two consecutive points $(a, j), (a + 1, j')$ in F^* , per construction, $M_\rho^\beta[\pi(a + 1), j'] = -1$. We add all points (x, j') with $x \in [\pi(a), \pi(b)]$. By Lemma 10, for all these points (x, j') it must be that $M_\rho^\beta[x, j'] < 1$. Thus, we found a walk F from $(1, 1)$ to (n, m) where for every $(i, j) \in F$, $M_\rho^\beta[i, j] < 1$ and the Fréchet distance between P and Q is at most $(1 + \alpha)(1 + \alpha + \beta)\rho$.

If no such path F^* exists. we claim that $D_{\mathcal{F}}(P, Q) > \rho$. Suppose for the sake of contradiction that $D_{\mathcal{F}}(P, Q) \leq \rho$ then there exists an xy -monotone path F from $(1, 1)$ to (n, m) where for all $(i, j) \in F$, $d(p_i, q_j) \leq \rho$. We use F to construct a path F^* from $(1, 1)$ to (n', m) in our grid graph. Specifically, for every element $(x, j) \in F$ we check if p_x has been removed during compression.

- If p_x has an equivalent in P^β then there exists an integer a such that $p_{\pi(a)} = p_x$ and we add the lattice point $(a, j) \in [n'] \times [m]$ to F^* . Per definition of F , $M_\rho^\beta[\pi(a), j] = -1$.
- Otherwise, we identify the index i such that $\pi(i)$ is the vertex of P^β preceding p_x and we add the point $(i, j) \in [n'] \times [m]$ to F^* . By Lemma 11, $M_\rho^\beta[\pi(i), j] < 1$.

Since F is a connected xy -monotone path from $(1, 1)$ to (n, m) , we obtain an xy -monotone path F^* from $(1, 1)$ to (n', m) . Moreover, whenever this path traverses a horizontal or diagonal edge to a point (a, j) it must be that $(\pi(a), j) \in F$ and thus $M_\rho^\beta[\pi(a), j] = -1$. Thus, F^* is a path from $(1, 1)$ to (n', m) in our grid graph which contradicts the earlier assumption that no such path exists. ◀

This corollary follows immediately from choosing $\alpha = \beta = 0.25(\sqrt{8\varepsilon + 9} - 3)$.

▶ **Corollary 14.** *Let G be a planar graph with N vertices, $P = (p_1, \dots, p_n)$ a κ -straight path and $Q = (q_1, \dots, q_m)$ be any walk in G . Given a value $\rho \in \mathbb{R}$ and some $\varepsilon > 0$ we correctly conclude either $D_{\mathcal{F}}(P, Q) > \rho$ or $D_{\mathcal{F}}(P, Q) \leq (1 + \varepsilon)\rho$ in $O(N \log N/\sqrt{\varepsilon} + n + \frac{\kappa}{\varepsilon}m)$ time.*

5 A conditional lower bound for computing the Fréchet distance

We show that for every $\delta > 0$ there is no $O((nm)^{1-\delta})$ algorithm for computing for the discrete Fréchet distance between two paths in a planar graph (unless OVH fails). We show this using a planar graph $G = (V, E)$ where the edges have integer weights in $\{0.001, 0.35, 0.6, 0.65, 1, 2, 3\}$.

36:12 On the Discrete Fréchet Distance in a Graph

In the full version we prove a similar statement for walks in a constant-complexity unit-weight graph. Throughout this section, we fix some $\delta > 0$ and $\gamma > 0$ and consider two sets A and B of d -dimensional Boolean vectors (with $d = \omega \log n$ where the constant ω depends on δ). In addition, we assume that A and B contain n' and m' vectors respectively with $n' = (m')^\gamma$. Using A and B , we reduce from Orthogonal Vectors using what we call a *vector gadget*. We construct a graph G and two paths P and Q where $D_{\mathcal{F}}(P, Q) < 3$ if and only if there exists $(a, b) \in A \times B$ such that a and b are orthogonal.

Proof notation. Throughout this section, we label vertices to represent an equivalence class. We construct a graph where we label “blue” vertices with a label in $\{x, y, z, B^{\{0\}}, B^{\{1\}}, B\}$ and “red” vertices with a label in $\{\alpha, \alpha^*, \beta, \beta^*, \gamma, A^{\{0\}}, A^{\{1\}}, A\}$. Ideally, we would construct a graph where for every red-blue pair of labels, all red-blue vertices with those two labels have the same distance. We maintain a slightly weaker property: consider any red-blue pair of vertices b, r with $\text{LABEL}(b) \in \{x, y, z, B^{\{0\}}, B^{\{1\}}, B\}$ and $\text{LABEL}(r) \in \{\alpha, \alpha^*, \beta, \beta^*, \gamma, A^{\{0\}}, A^{\{1\}}, A\}$. We demand the following: if $d(b, r) < 3$ then for all (b', r') with $\text{LABEL}(b') = \text{LABEL}(b)$ and $\text{LABEL}(r') = \text{LABEL}(r)$ it must be that $d(b', r') < 3$.

We construct for every vector in A (and B) a vector gadget. This gadget resembles the gadget used in the conditional lower bound for the Fréchet distance in the Euclidean plane by Bringmann [7]. The path P will traverse all vector gadgets of A in sequence (and Q will traverse gadgets of B). We connect all gadgets of A to all gadgets of B via “star” vertices (grey triangles or diamonds). These stars ensure that there can be a matching between every pair of gadgets (vectors). Finally, we add “park” vertices (square vertices) which are vertices of A (or B) that are close to all vertices of B (or A). The intuition is, that during a traversal (reparametrization) of P and Q an entity can remain stationary at a park vertex, whilst the other entity traverses their corresponding path until the appropriate gadgets can be matched.

Vector gadget. We illustrate the vector gadget for vectors $b \in B$ (see Figure 8). The “core” of this subgraph is vertex y connected to the following construction (repeated d times): there are two *Boolean vertices* $(B^{\{0\}}, B^{\{1\}})$, followed by an *intermediary* vertex B . This core will allow us to model a d -dimensional Boolean vector. We connect the core to two park vertices x and z where we add an edge (x, y) and (B, z) of weight 3. Finally, we add two star vertices where every vertex B, y and x get connected to the top star vertex, and every vertex $x, B^{\{0\}}, z$ get connected to the bottom star vertex. For every vector in A , the corresponding vector gadget is nearly identical. Most crucially, this subgraph is vertically mirrored and the edges attached to star vertices have different weights.

From gadgets to a graph. Given our instance of OV, we construct $(n + m)$ vector gadgets. Next, we combine the gadgets (Figure 9). We highlight the important steps: all the vector gadgets of B (and A) are placed horizontally adjacent to each other.

The vertices $\{s^\downarrow, z, \sigma^\uparrow\}$ get connected via a star vertex in the centre of the graph. Each vertex s^\uparrow gets connected to a star vertex at the top of the graph. Each vertex σ^\downarrow gets connected to a star vertex at the bottom of the graph. These two stars get connected via an edge with weight 2. Given this graph G , we say that a red vertex r is *close* to a blue vertex b if $d(r, b) < 3$. For every blue label, we observe the set of close red labels (Table 1):

Constructing the paths P and Q . Given G, A and B , we construct a path P consisting of $n = O(n' \cdot d)$ vertices and a path Q consisting of $m = O(m' \cdot d)$ vertices (refer to Figure 9). The path P starts in α and then moves to α^* . Then, P traverses every vector gadget of A in

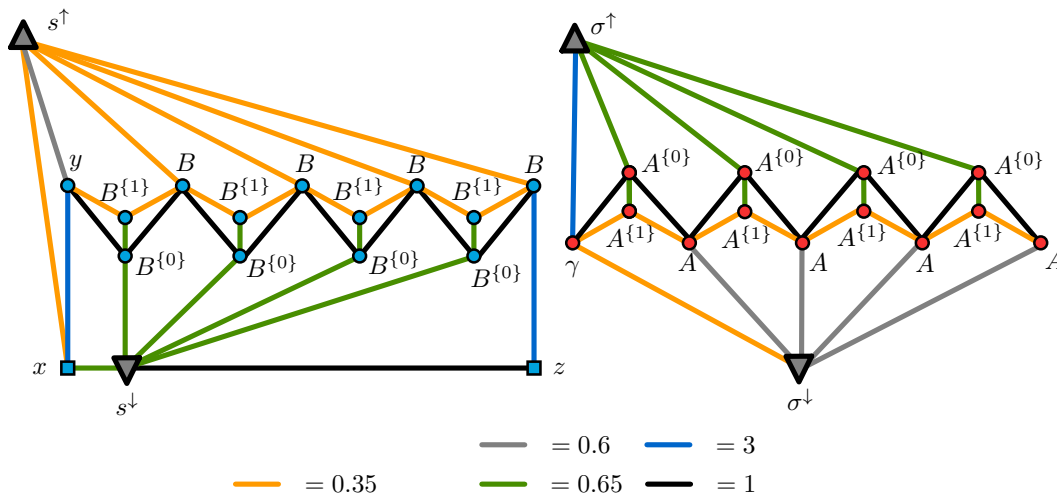


Figure 8 The gadgets for vectors in B and in A . The path corresponding to B will traverse blue vertices, the path corresponding to A red.

Table 1 The shortest distance between vertices with a label in $\{\alpha, \alpha^*, \beta, \beta^*, \gamma, A^{0}, A^{1}, A\}$ and in $\{x, y, z, B^{0}, B^{1}, B\}$, showing far and near pairs of labels.

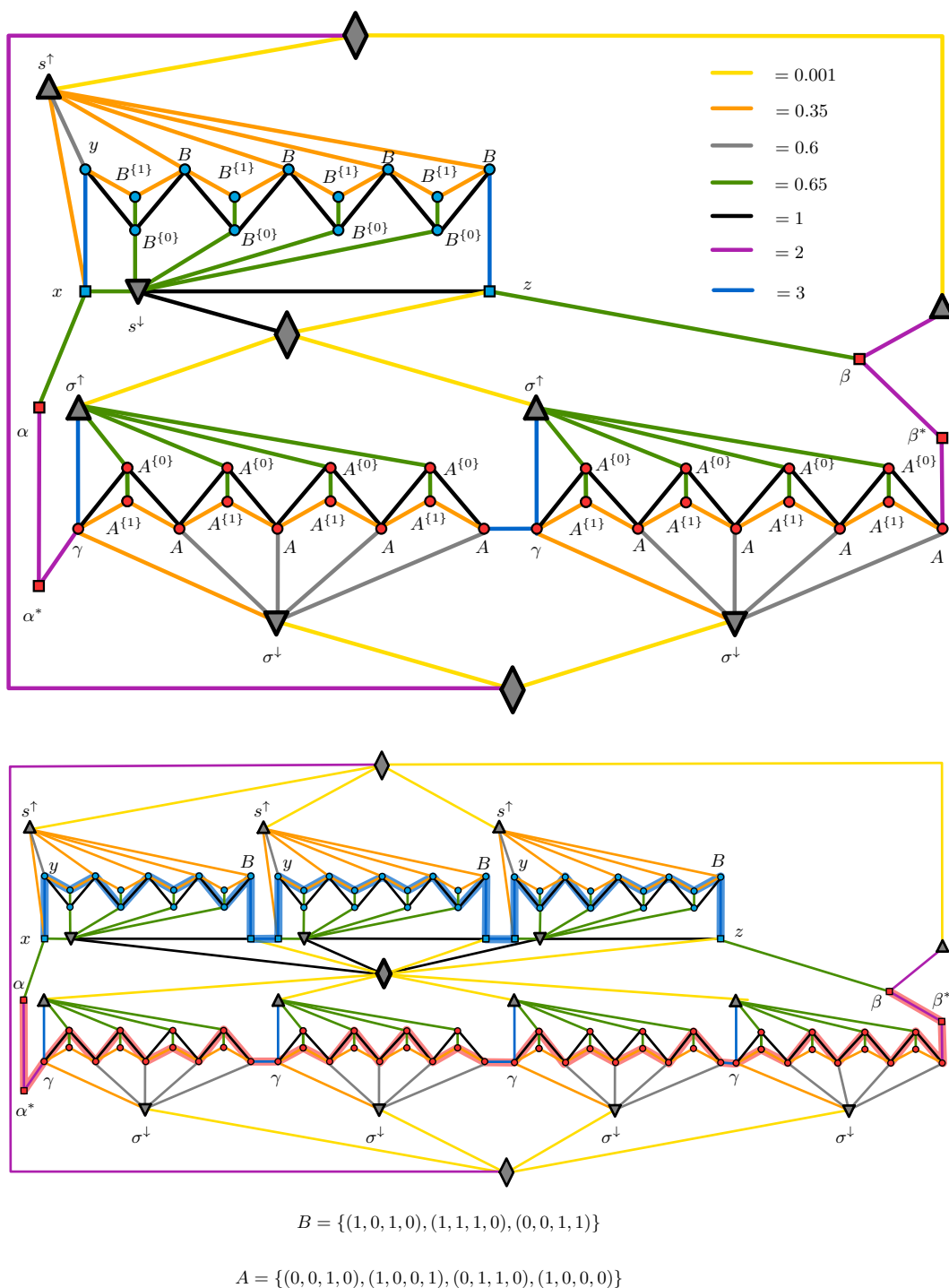
dist.	α	α^*	β	β^*	γ	A^{0}	A^{1}	A
x	.65	2.65	2.3	4.3	2.702	2.301	2.951	2.952
y	1.6	3.6	2.601	4.601	2.952	3.251	3.302	3.202
z	2.3	4.3	.65	2.65	1.652	0.652	1.302	1.652
B^{0}	1.95	3.95	2.3	4.3	3.301	2.301	2.951	3.301
B^{1}	1.7	3.7	2.701	4.701	3.051	2.951	3.402	3.302
B	1.35	3.35	2.351	4.351	2.702	3.001	3.051	2.951

sequence. Let v be the first vector in A . The path P arrives at y and traverses the Boolean vertices and intermediate vertices in an alternating manner (where P traverses A^{0} if the corresponding Boolean in v is false and A^{1} if the corresponding Boolean is true). Having traversed every vector gadget, P moves through β^* to β . The path Q traverses every vector gadget of B in sequence. Let a gadget correspond to a vector $v' \in B$:

The path Q starts at the vector x in the gadget and then traverses the Boolean vertices and intermediate vertices in an alternating manner (where Q traverses B^{0} if the corresponding Boolean in v' is false and B^{1} if the corresponding Boolean is true). The path Q ends at the vector z , and continues to the next gadget.

► **Theorem 15.** *Let G be a planar, integer-weighted graph, P and Q be two paths in G with n and m vertices and $n = m^\gamma$ for some constant $0 < \gamma \leq 1$. For all $\delta > 0$, there can be no algorithm that computes (a 1.01-approximation of $D_{\mathcal{F}}(P, Q)$) in $O((nm)^{1-\delta})$ time.*

Proof. For any given A and B of n' and m' vectors, we construct two paths P and G with $n = O(n' \log n')$ and $m = O(m' \log m')$ vertices respectively. OVH postulates that there exists no algorithm that can conclude if there exists two orthogonal vectors $(a, b) \in A \times B$ in $O((nm)^{1-\delta})$ time, for any $\delta > 0$. We prove this theorem by showing that there are two such vectors if and only if $D_{\mathcal{F}}(P, Q) < 3$. We observe that in our graph for all red/blue vertices r and b either $d(r, b) \leq 2.96$ or $d(r, b) \geq 3$ (which implies this proof for the 1.01-approximation).



■ **Figure 9** Top: we show how pairwise gadgets get connected. Bottom: given a set A of four and B of three vectors, we construct the corresponding graph and path.

We show that if there exist two orthogonal vectors $(a, b) \in A \times B$ then $D_{\mathcal{F}}(P, Q) < 3$. We construct a traversal of P and Q where the red entity (henceforth “Red”) traversing P remains close to the blue entity (“Blue”) traversing Q . First, Red is stationary at the park vertex α , whilst Blue traverses B until it reaches the vector gadget corresponding to $b \in B$. Then, whilst Blue remains stationary at the park vertex x , Red traverses P until it reaches the vector gadget corresponding to $a \in A$. At this point, Blue moves to y as Red moves to γ . Both entities simultaneously traverse their vector gadgets. During this traversal (since a and b are orthogonal) the entities remain close. Then, Blue remains stationary at z , whilst Red traverses the rest of P . Finally, Red remains at β whilst Blue traverses the rest of Q .

We show that if $D_{\mathcal{F}}(P, Q) < 3$ then there exists a pair of vectors $(a, b) \in A \times B$ such that a and b are orthogonal. Indeed, fix any traversal of P and Q that realises the Fréchet distance. When Red is at α^* , Blue must be at some vertex x .

Consider now the time when Blue moves from x to y (where y lies in a gadget corresponding to some vector $b \in B$). At this time, Red cannot be at the park vertex α because α precedes α^* . Similarly, Red cannot be at the park vertex β because β^* precedes β (and β^* is not close to x). Since $\text{CLOSE}(y) = \{\gamma, \alpha, \beta\}$, it must be that Blue is at some vertex γ (corresponding to some vector $a \in A$). Now consider the next time step, when we assume that Red moves to $\{A^{\{0\}}, A^{\{1\}}\}$ (the argument for when Blue moves to $\{B^{\{0\}}, B^{\{1\}}\}$ is symmetrical). If Red moves to $A^{\{0\}}$ then, via the same argument as above, Blue has to simultaneously move to $B^{\{0\}}$ or $B^{\{1\}}$. If Red moves to $A^{\{1\}}$ then Blue must move to $B^{\{0\}}$. For the next time step, via the same argument, both entities must move to A and B . We can continue this same argument, which shows that the two vectors a and b must be orthogonal. ◀

6 Concluding remarks

This paper is the first to study the natural question of computing the Fréchet distance between walks P and Q in graphs. Our algorithmic results (including the Voronoi diagram construction) do not depend on the planarity of G ; we rely only on a distance oracle. Hence, our result immediately holds for other classes of graphs where it is possible to efficiently construct distance oracles or in computational models where the distance oracle is provided. Given a distance oracle, our $(\kappa + 1)$ approximation is obtained in time (near-) linear in $(|P| + |Q|)$. In other words, our result in Section 3 allows us to pre-process a graph G in time nearly linear to its vertices, in order to efficiently facilitate Fréchet distance queries between two any two walks in (as long as one of the two walks is κ straight for some query constant κ). This is not true for our $(1 + \varepsilon)$ -approximation algorithm, which currently requires the construction of a Voronoi diagram of P in G and thus, for every pair of walks, must spend near-linear time in G .

References

- 1 Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 434–443. IEEE, 2014.
- 2 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995.
- 3 Helmut Alt, Christian Knauer, and Carola Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38(1):45–58, 2004.

- 4 Boris Aronov, Sarel Har-Peled, Christian Knauer, Yusu Wang, and Carola Wenk. Fréchet distance for curves, revisited. In *European symposium on algorithms*, pages 52–63. Springer, 2006.
- 5 Alessandro Bombelli, Lluís Soler, Eric Trumbauer, and Kenneth D Mease. Strategic air traffic planning with Fréchet distance aggregation and rerouting. *Journal of Guidance, Control, and Dynamics*, 40(5):1117–1129, 2017.
- 6 Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *Proceedings of the 31st international conference on Very large data bases*, pages 853–864, 2005.
- 7 Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly sub-quadratic algorithms unless Seth fails. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 661–670. IEEE, 2014.
- 8 Karl Bringmann and Marvin Künnemann. Improved approximation for Fréchet distance on c -packed curves matching conditional lower bounds. *International Journal of Computational Geometry & Applications*, 27(01n02):85–119, 2017.
- 9 Karl Bringmann and Wolfgang Mulzer. Approximability of the discrete Fréchet distance. *Journal of Computational Geometry*, 7(2):46–76, 2016.
- 10 Kevin Buchin, Maike Buchin, David Duran, Brittany Terese Fasy, Roel Jacobs, Vera Sacristan, Rodrigo I Silveira, Frank Staals, and Carola Wenk. Clustering trajectories for map construction. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10, 2017.
- 11 Kevin Buchin, Maike Buchin, Wouter Meulemans, and Wolfgang Mulzer. Four Soviets walk the dog: Improved bounds for computing the Fréchet distance. *Discrete & Computational Geometry*, 58(1):180–216, 2017.
- 12 Kevin Buchin, Maike Buchin, and Yusu Wang. Exact algorithms for partial curve matching via the Fréchet distance. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 645–654. SIAM, 2009.
- 13 Kevin Buchin, Tim Ophelders, and Bettina Speckmann. Seth says: Weak Fréchet distance is faster, but only if it is continuous and in one dimension. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2887–2901. SIAM, 2019.
- 14 Maike Buchin, Bernhard Kilgus, and Andrea Kölzsch. Group diagrams for representing trajectories. *International Journal of Geographical Information Science*, 34(12):2401–2433, 2020.
- 15 Erin Wolf Chambers, Eric Colin De Verdiere, Jeff Erickson, Sylvain Lazard, Francis Lazarus, and Shripad Thite. Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry*, 43(3):295–311, 2010.
- 16 Timothy M Chan and Zahed Rahmati. An improved approximation algorithm for the discrete Fréchet distance. *Information Processing Letters*, 138:72–74, 2018.
- 17 Panagiotis Charalampopoulos, Paweł Gawrychowski, Shay Mozes, and Oren Weimann. Almost optimal distance oracles for planar graphs. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23–26, 2019*, pages 138–151. ACM, 2019. doi:10.1145/3313276.3316316.
- 18 Vincent Cohen-Addad, Søren Dahlgaard, and Christian Wulff-Nilsen. Fast and compact exact distance oracle for planar graphs. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15–17, 2017*, pages 962–973. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.93.
- 19 Connor Colombe and Kyle Fox. Approximating the (continuous) Fréchet distance. In *37th International Symposium on Computational Geometry (SoCG 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- 20 Thomas Devogele. A new merging process for data integration based on the discrete Fréchet distance. In *Advances in spatial data handling*, pages 167–181. Springer, 2002.


- 21 Anne Driemel and Sarel Har-Peled. Jaywalking your dog: computing the Fréchet distance with shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013.
- 22 Anne Driemel, Sarel Har-Peled, and Carola Wenk. Approximating the Fréchet distance for realistic curves in near linear time. *Discret. Comput. Geom.*, 48(1):94–127, 2012. doi:10.1007/s00454-012-9402-z.
- 23 Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 1994.
- 24 David Göckede. Computing the Fréchet distance in graphs efficiently using shortest-path distance oracles. Master’s thesis, Department of Computer Science, University of Bonn, 2021.
- 25 Qian-Ping Gu and Gengchun Xu. Constant query time $(1+\varepsilon)$ -approximate distance oracle for planar graphs. *Theor. Comput. Sci.*, 761:78–88, 2019. doi:10.1016/j.tcs.2018.08.024.
- 26 Joachim Gudmundsson, Majid Mirzanezhad, Ali Mohades, and Carola Wenk. Fast Fréchet distance between curves with long edges. *International Journal of Computational Geometry & Applications*, 29(02):161–187, 2019.
- 27 Atlas F Cook IV and Carola Wenk. Geodesic Fréchet distance inside a simple polygon. *ACM Transactions on Algorithms (TALG)*, 7(1):1–19, 2010.
- 28 Minghui Jiang, Ying Xu, and Binhai Zhu. Protein structure–structure alignment with discrete Fréchet distance. *Journal of bioinformatics and computational biology*, 6(01):51–64, 2008.
- 29 Philip N. Klein. Preprocessing an undirected planar network to enable fast approximate distance queries. In David Eppstein, editor, *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA*, pages 820–827. ACM/SIAM, 2002. URL: <http://dl.acm.org/citation.cfm?id=545381.545488>.
- 30 Philip N. Klein. Multiple-source shortest paths in planar graphs. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 146–155. SIAM, 2005. URL: <http://dl.acm.org/citation.cfm?id=1070432.1070454>.
- 31 Maximilian Konzack, Thomas McKetterick, Tim Ophelders, Maike Buchin, Luca Giuggioli, Jed Long, Trisalyn Nelson, Michel A Westenberg, and Kevin Buchin. Visual analytics of delays and interaction in movement data. *International Journal of Geographical Information Science*, 31(2):320–345, 2017.
- 32 Yaowei Long and Seth Pettie. Planar distance oracles with better time-space tradeoffs. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2517–2537. SIAM, 2021. doi:10.1137/1.9781611976465.149.
- 33 Anil Maheshwari, Jörg-Rüdiger Sack, Kaveh Shahbaz, and Hamid Zarrabi-Zadeh. Fréchet distance with speed limits. *Computational Geometry*, 44(2):110–120, 2011.
- 34 Ariane Mascaret, Thomas Devogele, Iwan Le Berre, and Alain Hénaff. Coastline matching process based on the discrete Fréchet distance. In *Progress in Spatial Data Handling*, pages 383–400. Springer, 2006.
- 35 Jiri Matousek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2013.
- 36 Liam Roditty, Mikkel Thorup, and Uri Zwick. Deterministic constructions of approximate distance oracles and spanners. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, volume 3580 of *Lecture Notes in Computer Science*, pages 261–272. Springer, 2005. doi:10.1007/11523468_22.
- 37 Roniel S. De Sousa, Azzedine Boukerche, and Antonio A. F. Loureiro. Vehicle trajectory similarity: Models, methods, and applications. *ACM Comput. Surv.*, 53(5), September 2020. doi:10.1145/3406096.

- 38 E Sriraghavendra, K Karthik, and Chiranjib Bhattacharyya. Fréchet distance based approach for searching online handwritten documents. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, pages 461–465. IEEE, 2007.
- 39 Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal*, 29(1):3–32, 2020.
- 40 Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM (JACM)*, 51(6):993–1024, 2004.
- 41 Mikkel Thorup and Uri Zwick. Approximate distance oracles. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 183–192. ACM, 2001. doi:10.1145/380752.380798.
- 42 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005. doi:10.1016/j.tcs.2005.09.023.
- 43 Christian Wulff-Nilsen. Approximate distance oracles with improved query time. In *Encyclopedia of Algorithms*, pages 94–97. Springer, 2016. doi:10.1007/978-1-4939-2864-4_568.
- 44 Dong Xie, Feifei Li, and Jeff M Phillips. Distributed trajectory similarity search. *Proceedings of the VLDB Endowment*, 10(11):1478–1489, 2017.

Computing a Link Diagram from Its Exterior

Nathan M. Dunfield   

Dept. of Math., University of Illinois at Urbana-Champaign, IL, USA

Malik Obeidin 

Google, Inc., Mountain View, CA, USA

Cameron Gates Rudd   

Dept. of Math., University of Illinois at Urbana-Champaign, IL, USA

Abstract

A knot is a circle piecewise-linearly embedded into the 3-sphere. The topology of a knot is intimately related to that of its exterior, which is the complement of an open regular neighborhood of the knot. Knots are typically encoded by planar diagrams, whereas their exteriors, which are compact 3-manifolds with torus boundary, are encoded by triangulations. Here, we give the first practical algorithm for finding a diagram of a knot given a triangulation of its exterior. Our method applies to links as well as knots, and allows us to recover links with hundreds of crossings. We use it to find the first diagrams known for 23 principal congruence arithmetic link exteriors; the largest has over 2,500 crossings. Other applications include finding pairs of knots with the same 0-surgery, which relates to questions about slice knots and the smooth 4D Poincaré conjecture.

2012 ACM Subject Classification Mathematics of computing → Geometric topology

Keywords and phrases computational topology, low-dimensional topology, knot, knot exterior, knot diagram, link, link exterior, link diagram

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.37

Related Version *Full Version*: <https://arxiv.org/abs/2112.03251v2>

Supplementary Material *Software (Source Code and Data)*: <https://doi.org/10.7910/DVN/BT1M8R>

Funding *Nathan M. Dunfield*: Partially supported by US National Science Foundation grants DMS-1510204 and DMS-1811156 and by a Simons Fellowship.

Malik Obeidin: Partially supported by US National Science Foundation grants DMS-1510204 and DMS-181115.

Cameron Gates Rudd: Partially supported by US National Science Foundation grant DMS-1811156.

Acknowledgements We thank Matthias Goerner and Henry Segerman for helpful correspondence, and thank the referees for their detailed comments which helped improve this paper.

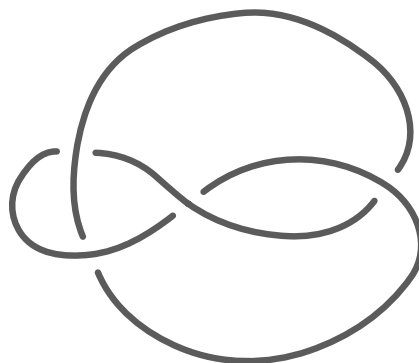


Figure 1 A planar diagram for a knot is a 4-valent graph with a planar embedding where every vertex represents a *crossing*, a place where one part of the knot crosses in front of the other in 3D.



© Nathan M. Dunfield, Malik Obeidin, and Cameron Gates Rudd;
licensed under Creative Commons License CC-BY 4.0

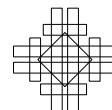
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 37; pp. 37:1–37:24

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

A knot is a piecewise-linear (PL) embedding of a circle S^1 into the 3-sphere S^3 . The study of knots goes back to the 19th century, and today is a central focus of low-dimensional topology, with applications to chemistry [23], biology [24], engineering [40], and theoretical computer science [16]. Two knots are topologically equivalent when they are isotopic, that is, when one can be continuously deformed to the other without passing through itself. Computationally, knots are typically encoded as planar diagrams (Figure 1); there are more than 350 million distinct knots with diagrams of at most 19 crossings as enumerated by [11].

The topology of knots is intimately related to that of their exteriors, where the *exterior* of a knot K is the compact 3-manifold with torus boundary $E(K) := S^3 \setminus N(K)$ where $N(K)$ is an open tubular neighborhood of K . Indeed, the exterior $E(K)$ determines the knot K [26]. Many algorithms for knots work via their exteriors, starting with Haken’s foundational method for deciding when a knot is equivalent to a round circle [27]. Consequently, the problem of going from a diagram D of K to a triangulation of $E(K)$ is well-studied [28, §7]; for ideal triangulations (see Section 2.1 below), one needs only four tetrahedra per crossing of D [51, §3]. Here, we study the inverse problem:

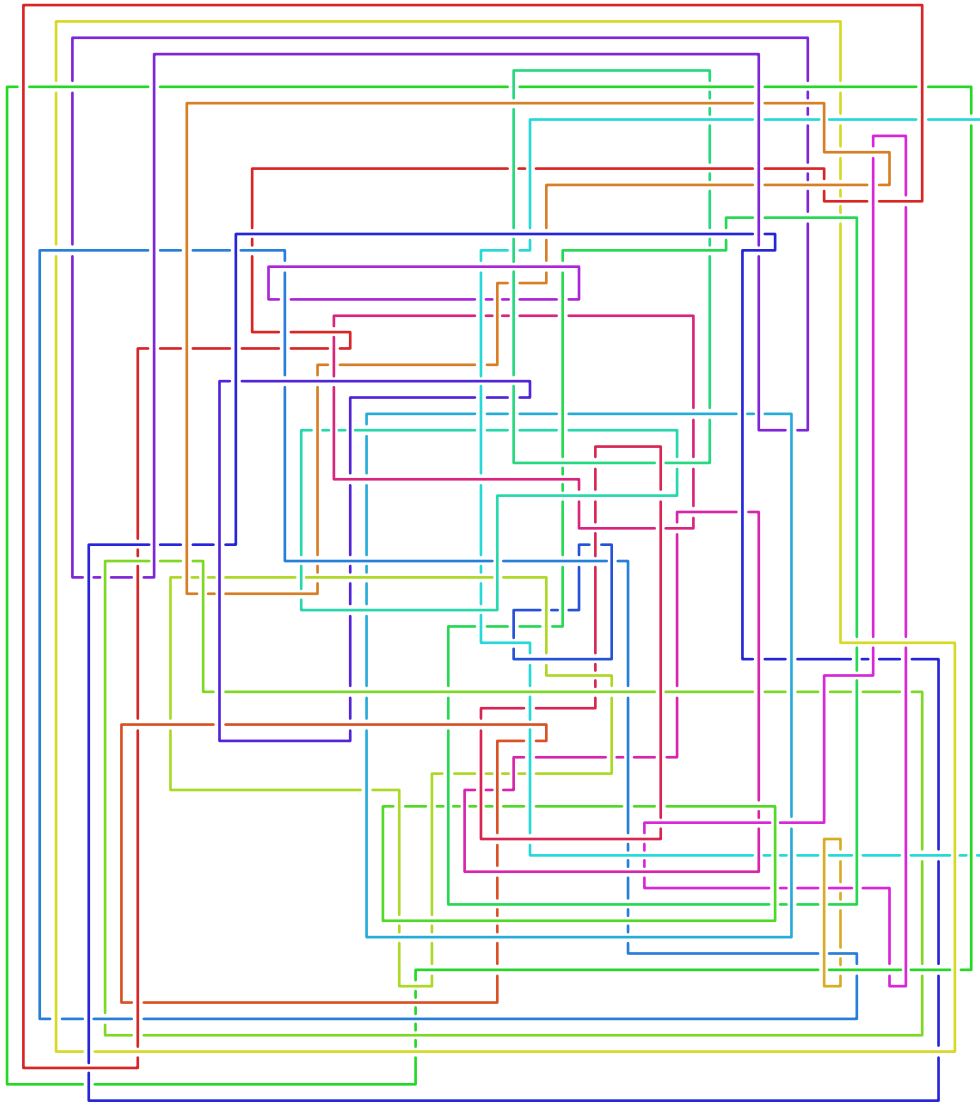
► **Find Diagram.** *Input a triangulation \mathcal{T} of a knot exterior $E(K)$, output a diagram of K .*

If the input triangulation \mathcal{T} is guaranteed to be that of a knot exterior (in fact, this is decidable by Algorithm S of [33]), then a useless algorithm to find D is just this: start generating all knot diagrams, triangulate each exterior, and then do Pachner moves (see Section 2.4) on these triangulations. Since any two triangulations of a compact 3-manifold are connected by a sequence of such moves, one eventually stumbles across \mathcal{T} , thus finding a diagram for the underlying knot. We do not explore the computational complexity of FIND DIAGRAM here (though it is at least exponential space by Theorem E.1 in Appendix E of the full version [18]), but rather give the first algorithm that is highly effective in practice. We work more generally with links, where a *link* is a disjoint union of knots. While a link exterior does not uniquely determine a link [3, Figure 9.28], this indeterminacy is removed by specifying meridional curves for the link; hence we require such curves as part of the input in Section 1.2. Figures 2 and 3 show diagrams that were found by our method; these are the first known diagrams of these particular link exteriors, see Section 9.1.

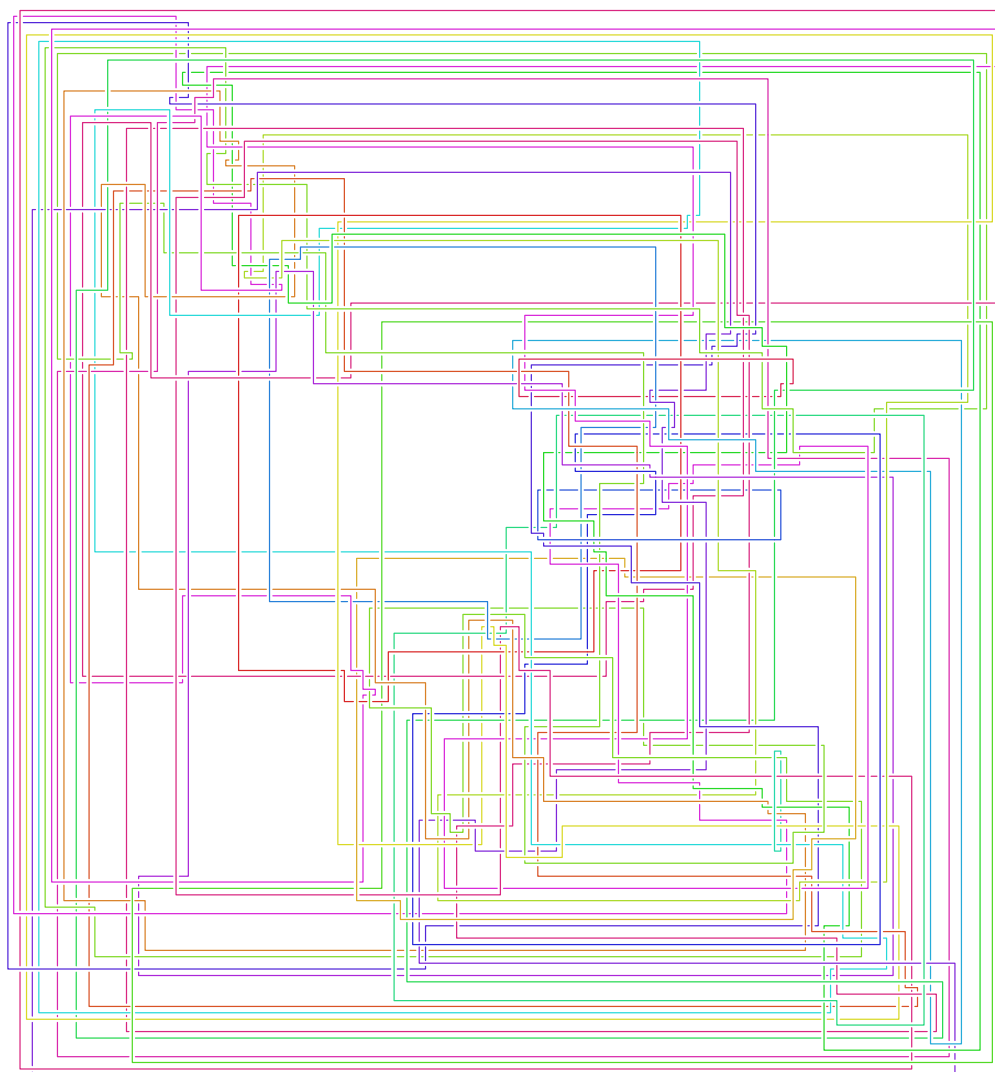
1.1 Prior work

The case when the interior of $E(K)$ has a complete hyperbolic structure, in short is *hyperbolic*, is in practice generic for prime knots; for example, 99.999% of the knots in [11] are hyperbolic. The homeomorphism problem for such 3-manifolds can be quickly solved in practice using hyperbolic geometry, even for triangulations with 1,000 tetrahedra [53]. This allows a table lookup method for FIND DIAGRAM when K is small enough; one uses hyperbolic and homological invariants to form a hash of $E(K)$, queries a database of knots to get a handful of possible K_i , and then checks if any $E(K_i)$ is homeomorphic to $E(K)$. This technique is used by the `identify` method of [15], but is hopeless for something like Figure 2, as the number of links of that size exceeds the number of atoms in the visible universe [47].

A related approach was used in [14, 5] to find knot diagrams for all 1,267 knots where $E(K)$ is hyperbolic and can be triangulated with at most 9 ideal tetrahedra [8, 17]. While knots with few crossings have simple exteriors, the converse is not the case, and the simplest known diagrams for about 25% of these knots have 100–300 crossings. However, these knots either fall into very special families which can be tabulated to a large number of crossings, or



■ **Figure 2** The first known diagram of a link whose exterior is $\dot{M} = \mathbb{H}^3/\Gamma(I)$ where $\Gamma(I)$ is the principal congruence subgroup of $\mathrm{PSL}_2\mathbb{Z}[\frac{1+\sqrt{15}i}{2}]$ of level $I = \langle 6, -3+\sqrt{15}i \rangle$ from [6]; it has 24 components and 294 crossings. The input ideal triangulation $\dot{\mathcal{T}}$ for \dot{M} had 249 tetrahedra. Since the hyperbolic volume of $\dot{M} \approx 225.98$, any diagram must have at least 66 crossings by [1, Theorem 5.1].



■ **Figure 3** The first known diagram of a link whose exterior is $\dot{M} = \mathbb{H}^3/\Gamma(I)$ where $\Gamma(I)$ is the principal congruence subgroup of $\mathrm{PSL}_2\mathbb{Z}[\frac{1+\sqrt{15}i}{2}]$ of level $I = \langle 5, \frac{5+\sqrt{15}i}{2} \rangle$ from [6]; it has 24 components and 1,092 crossings. The input ideal triangulation $\dot{\mathcal{T}}$ for \dot{M} had 211 tetrahedra. Since the hyperbolic volume of $\dot{M} \approx 188.32$, any diagram must have at least 56 crossings [1, Theorem 5.1].

one can drill out additional curves to get a link exterior that appears in an existing table and has special properties allowing the recovery of a diagram of the knot itself. There are other ad hoc methods in the literature, see e.g. [7] and references therein, but this paper is the first to give a generically applicable method for FIND DIAGRAM.

1.2 Outline of algorithm

As Figures 2 and 3 show, our method can solve FIND DIAGRAM in some cases where any diagram for the link has 55 or more crossings. We also easily recover everything in Section 1.1, and more applications are given in Sections 8 and 9. Experimental mean running time was $O(1.07^n)$, see Figure 14. With the definitions of Section 2, the input for our algorithm is:

► Input.

- a. An ideal triangulation \mathcal{T} of a compact 3-manifold \mathring{M} with toroidal boundary, with an essential simple closed curve α_i for each boundary component of \mathring{M} .
- b. A sequence (P_i) of Pachner moves transforming the layered filling triangulation \mathcal{T} of the manifold $M = \mathring{M}(\alpha_1, \dots, \alpha_k)$ into a specific 2-tetrahedra *base triangulation* \mathcal{T}_0 of S^3 .

You might object that (b) is effectively cheating, since no polynomial-time algorithm for finding (P_i) is known, or indeed for deciding if M is S^3 . Using the estimates in [37], one can perform a naive search to find some (P_i) , but the complexity of this is super-exponential. However, recognizing S^3 by finding such moves is easy in practice, see Section 7, with the length of (P_i) linear in the size of \mathcal{T} as per Figure 16. The output of the algorithm is a knot diagram D , encoded as a planar graph with over/under crossing data for the vertices.

The main data structure is a triangulation \mathcal{T} of S^3 with a PL link L that is disjoint from the 1-skeleton. The link L is encoded as a sequence of line segments, each contained in a single tetrahedron of \mathcal{T} , with endpoints recorded in barycentric coordinates. An initial pair (\mathcal{T}, L) in (b) is constructed from input (a) as described in Appendix A of the full version [18]. The algorithm proceeds by performing the Pachner moves P_i from (b), keeping track of the PL arcs encoding the link L throughout using the techniques of Section 3. The result is the base triangulation enriched with PL arcs representing the link L . As detailed in Section 5, this triangulation of S^3 can be cut open along faces and embedded in \mathbb{R}^3 , giving an embedding of the cut-open link into \mathbb{R}^3 as a collection of PL arcs with endpoints on the boundary of these tetrahedra. As in Figure 11, these PL arcs are then tied up using the face identifications to obtain a collection of closed PL curves that represent L . An initial link diagram D is obtained by projecting this PL link onto a plane and recording crossing information. We then apply generic simplification methods to D and output the result.

This outline turns out to be deceptively simple. Some key difficulties are:

1. Understanding what $2 \rightarrow 3$ and $3 \rightarrow 2$ Pachner moves do to the link L is fairly straightforward as these correspond to changing the triangulation of a convex polyhedron in \mathbb{R}^3 . However, while these two moves theoretically suffice for (b), in practice one wants to use $2 \rightarrow 0$ moves as well, see Section 7, and these are much harder to deal with, as Figure 6 shows. We thus expand each $2 \rightarrow 0$ move into a (sometimes quite lengthy) sequence of $2 \rightarrow 3$ and $3 \rightarrow 2$ moves as discussed in Section 4. We give a simplified expansion for the trickiest part, the endpoint-through-endpoint move, using 6 of the basic $2 \rightarrow 3$ and $3 \rightarrow 2$ moves instead of 14.
2. The complexity of the link grows very rapidly as we do Pachner moves, resulting in enormously complicated initial diagrams. We greatly reduce this by elementary local simplifications to the link after each Pachner move, see Section 3.2.
3. Prior work on simplifying link diagrams was focused on those with 30 or fewer crossings, where random application of Reidemeister moves (plus flypes) are extremely effective. Here, we need to simplify diagrams with 10,000 or even 100,000 crossings down to something with less than 100, and such methods proved ineffective for this. Instead, we used the more global *strand pickup* method of Section 6.

2 Background

2.1 Triangulations

Let M be a compact orientable 3-manifold, possibly with boundary. A *triangulation* of M is a cell complex \mathcal{T} made from finitely many tetrahedra by gluing some of their 2-dimensional faces in pairs via orientation-reversing affine maps so that the resulting space is homeomorphic

to M . These triangulations are not necessarily simplicial complexes, but rather what are sometimes called semi-simplicial, pseudo-simplicial, or singular triangulations. Of particular importance are those with a single vertex, the *1-vertex triangulations*. We use \mathcal{T}^i to denote the i -skeleton of \mathcal{T} . When M has nonempty boundary, an *ideal triangulation* of M is a cell complex \mathcal{T} made out of finitely many tetrahedra by gluing *all* of their 2-dimensional faces in pairs as above so that $M \setminus \partial M$ is homeomorphic to $\mathcal{T} \setminus \mathcal{T}^0$. Put another way, the manifold M is what you get by gluing together *truncated* tetrahedra in the corresponding pattern. See [49] for background on ideal triangulations, which we use only for 3-manifolds whose boundary is a union of tori. We always include the modifier “ideal”, so throughout “triangulation” means a non-ideal, also called “finite”, triangulation.

2.2 Triangulations with PL curves

Consider a tetrahedron Δ in \mathbb{R}^n as the convex hull of its vertices v_0, v_1, v_2 , and v_3 . We encode points in Δ using *barycentric coordinates*, that is, write $p \in \Delta$ as the unique convex combination $\sum_i x_i v_i$ and then represent p by the vector (x_0, x_1, x_2, x_3) , where of necessity $\sum_i x_i = 1$. For a 3-manifold triangulation \mathcal{T} , we view each tetrahedron τ as having a fixed identification with the tetrahedron in \mathbb{R}^4 whose vertices are the standard basis vectors; we use this to encode points in τ by barycentric coordinates.

An oriented PL curve in \mathcal{T} will be described by a sequence of such barycentric coordinates as follows. A *barycentric arc* a is an ordered pair of points (u, v) in a tetrahedron τ , representing the straight segment joining them. We write $a.\text{start} = u$ and $a.\text{end} = v$. A *barycentric curve* C is a sequence of barycentric arcs a_i such that $a_i.\text{end}$ and $a_{i+1}.\text{start}$ correspond to the same point in M under the face identifications of \mathcal{T} . For a barycentric curve, we define $a_i.\text{next} = a_{i+1}$ and $a_{i+1}.\text{past} = a_i$; these may not lie in the same tetrahedron. Suppose the barycentric curve C consists of N barycentric arcs. If $a_0.\text{start}$ and $a_N.\text{end}$ correspond to the same point in M , we have a *barycentric loop*. An embedded barycentric loop is a *barycentric knot*. A *barycentric link* is a finite disjoint union of such knots.

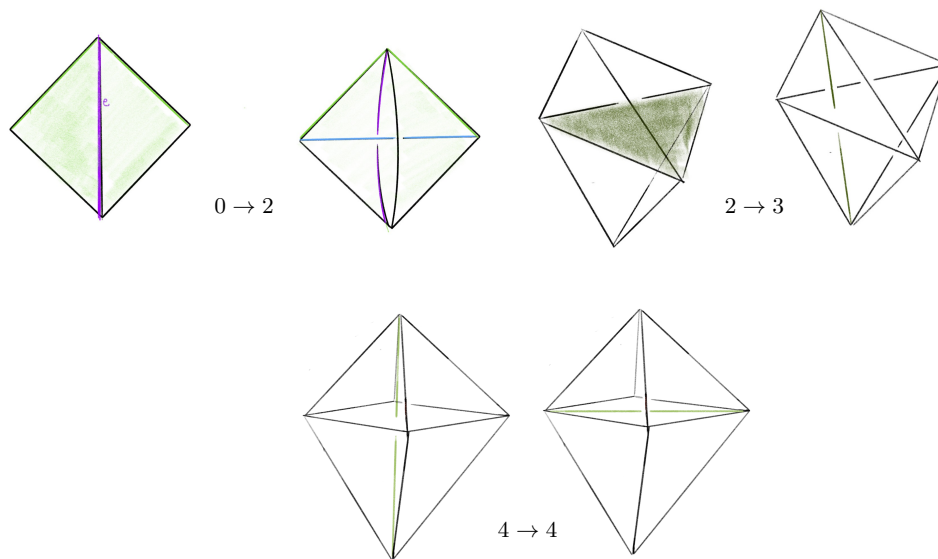
We always require that a barycentric curve C is in the following kind of general position with respect to \mathcal{T} . First, C is disjoint from \mathcal{T}^1 . Second, any intersection of a constituent barycentric arc a with \mathcal{T}^2 is an endpoint of a . Finally, arcs do not bounce off faces of \mathcal{T}^2 , so if an arc ends in a face, the next arc must be in the adjacent tetrahedron on the other side of that face. Throughout, we use only points whose barycentric coordinates are in \mathbb{Q} .

2.3 Dehn filling

Suppose \mathring{M} is a compact 3-manifold whose boundary is a union of tori. Given an essential simple closed curve α_i on each boundary component T_i , the *Dehn filling* of \mathring{M} along $\alpha = (\alpha_1, \dots, \alpha_k)$ is the closed 3-manifold $\mathring{M}(\alpha)$ obtained from \mathring{M} by gluing a solid torus $D^2 \times S^1$ to each T_i so that $\partial D^2 \times \{\text{point}\}$ is α_i . When \mathring{M} is the exterior of a link L in S^3 and each α_i is a small meridional loop about the i -th component of L , then $\mathring{M}(\alpha)$ is just S^3 . Given an ideal triangulation $\mathring{\mathcal{T}}$ of \mathring{M} and Dehn filling curves α , we follow [52, 32, 33] to create a 1-vertex triangulation \mathcal{T} of $\mathring{M}(\alpha)$ that we call the *layered filling triangulation*; see Appendix A of the full version [18]. A key point is that the link L consisting of the cores of the k added solid tori is a barycentric link in \mathcal{T} made of just k barycentric arcs.

2.4 Pachner moves

A 3-manifold triangulation \mathcal{T} can be modified by local *Pachner moves* (bistellar flips) to give a new triangulation of the same underlying manifold. Those we use are:



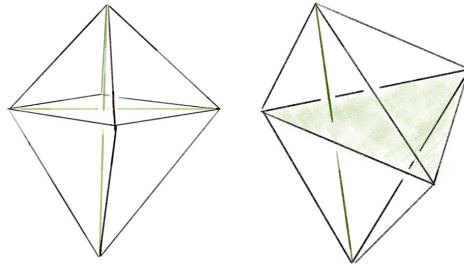
■ **Figure 4** Pachner moves which preserve the number of vertices.

1. The $2 \rightarrow 3$ move and its inverse $3 \rightarrow 2$ move. These take a triangulation of a ball, possibly with boundary faces glued together, and retriangulate the interior without changing the boundary triangulation. Specifically, the $2 \rightarrow 3$ move takes a pair of distinct tetrahedra sharing a face and replaces them with three new tetrahedra around a new central edge. The $3 \rightarrow 2$ move reverses this, replacing three distinct tetrahedra around a valence-3 edge with two tetrahedra sharing a face.
2. The $4 \rightarrow 4$ move. The $4 \rightarrow 4$ move takes four tetrahedra around a central edge and replaces them with four new tetrahedra assembled around a new valence-4 edge.
3. The $2 \rightarrow 0$ move and its inverse $0 \rightarrow 2$ move. The $2 \rightarrow 0$ move takes a pair of tetrahedra sharing two faces to form a valence-2 edge and collapses them onto their common faces. The $0 \rightarrow 2$ move reverses this by puffing air into a pair of faces sharing an edge and adding two new tetrahedra. We call the complex created by the $0 \rightarrow 2$ move a *pillow*. The $0 \rightarrow 2$ move inflates a pillow and the $2 \rightarrow 0$ move collapses a pillow.

If \mathcal{S} and \mathcal{T} are two 1-vertex triangulations of the same closed 3-manifold M , then there is a sequence of Pachner moves that transforms \mathcal{S} into \mathcal{T} ; provided both \mathcal{S} and \mathcal{T} have at least two tetrahedra, one needs only use $2 \rightarrow 3$ and $3 \rightarrow 2$ moves by [36, Theorem 1.2.5] (see also [39, 42]). When M is S^3 , any triangulation \mathcal{T} with n tetrahedra is related to a standard triangulation by at most $12 \cdot 10^6 n^2 2^{2 \cdot 10^3 n^2}$ Pachner moves [37]. Experimentally, one needs many fewer moves [9]. In our data shown in Figure 16, the number is $O(n)$; this is essential for the utility of our algorithm for FIND DIAGRAM.

3 Modifying triangulations with arcs

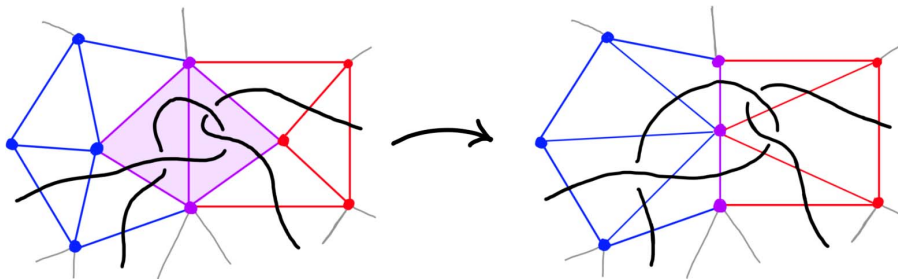
Using part (a) of the input data, we first build the layered filled triangulation \mathcal{T} of Section 2.3, which comes enriched with a barycentric link L . Part (b) of the input data is a sequence of Pachner moves (P_i) converting \mathcal{T} to the base triangulation \mathcal{T}_0 of Section 5. The next step of our algorithm is to apply the moves (P_i) to \mathcal{T} , carrying the link L along as we go.



■ **Figure 5** Two bipyramids with superimposed triangulations corresponding to before and after applying the $4 \rightarrow 4$ move and $2 \rightarrow 3$ or $3 \rightarrow 2$ moves.

3.1 Pachner moves with arcs

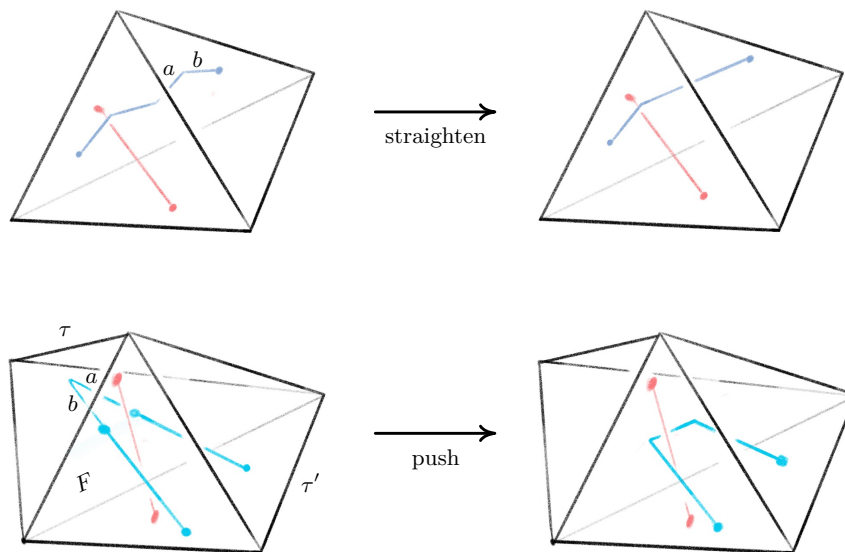
We call the $2 \rightarrow 3$, $3 \rightarrow 2$, and $4 \rightarrow 4$ moves the *simple Pachner moves*. Each simple Pachner move P takes a triangulated ball B in \mathcal{T} , possibly with boundary faces glued together, and re-triangulates B without changing the triangulation of ∂B to obtain $P\mathcal{T}$. The arcs of the link L contained in the ball are initially encoded using the barycentric coordinates of \mathcal{T} , and we need to re-express these arcs in the new barycentric coordinate system of $P\mathcal{T}$. We model each simple Pachner move as a pair of triangulations of concrete bipyramids in \mathbb{R}^3 , as shown in Figure 5. We identify the tetrahedra in \mathcal{T} and $P\mathcal{T}$ involved in P with tetrahedra in the corresponding bipyramid in \mathbb{R}^3 . This identification allows us to map barycentric arcs from \mathcal{T} into \mathbb{R}^3 , and then to map these arcs in \mathbb{R}^3 into $P\mathcal{T}$. Appendix B of the full version [18] details how this is used to give a method `with_arcs[P]` that applies a simple Pachner move P to \mathcal{T} while transferring the barycentric arcs from \mathcal{T} to $P\mathcal{T}$. This approach cannot work for the $2 \rightarrow 0$ move, as demonstrated by Figure 6. To implement `with_arcs[2 \rightarrow 0]`, we factor the $2 \rightarrow 0$ move into a sequence of $2 \rightarrow 3$ and $3 \rightarrow 2$ moves as described in Section 4.



■ **Figure 6** Cartoon showing the difficulty of doing a $2 \rightarrow 0$ move with arcs present. At left, the two tetrahedra in the pillow to be collapsed are shaded. Here, you should regard the vertical purple arc as the valence-2 edge, with the blue and red dots opposite being cross-sections of the two edges of the pillow that become identified in the collapse. The problem is that we have to push all the topology of the *link* out of the pillow before we collapse it, requiring us to move arcs into many of the tetrahedra adjacent to the pillow.

3.2 Simplifying arcs

Given the inputs (a) and (b) of Section 1.2, the machinery of Section 3.1 always produces the desired link L in the base triangulation \mathcal{T}_0 . However, even in the smallest examples, applying the sequence of Pachner moves to \mathcal{T} produces incredibly complicated configurations of arcs in \mathcal{T}_0 encoding L . This complexity makes necessary computational geometry tasks



■ **Figure 7** The *straighten* move removes unnecessary bends in the link, and the *push* move reduces unnecessary intersections with the 2-skeleton.

prohibitively expensive. Fortunately, much of this complexity is not topologically essential, and the number of arcs can be decreased dramatically by the basic simplifications we now describe. Without these, applying our full algorithm to an ideal triangulation $\tilde{\mathcal{T}}$ with just two tetrahedra resulted in 838 arcs and an initial link diagram with 5,130 crossings; with the simplifications, we get 19 arcs and 35 crossings. A 3-tetrahedra ideal triangulation resulted in 129,265 arcs compared to 27 with simplifications, and something with 10 tetrahedra would be impossible without them. Our two kinds of simplification moves are shown in Figure 7.

The first is **straighten**, which takes as input a tetrahedron τ with barycentric arcs. It then checks for each arc a in τ if the pair of arcs a and $b = a.\text{next}$ can be replaced with a single arc that runs from $a.\text{start}$ and $b.\text{end}$. The check is that no other arc in τ has an interior intersection with the triangle spanned by a and b . The other move is **push**, which removes unnecessary intersections with \mathcal{T}^2 . When a starts on the same face F that $b = a.\text{next}$ ends on, it checks whether any other arc intersects the triangle a and b span. If there are none, the move replaces a and b with an arc in the tetrahedron τ' glued to τ along F . This often produces a bend that can then be removed by a straighten move.

3.3 Computational geometry issues

Our algorithm requires many geometric computations with barycentric arcs, e.g. to test for one of our simplifying moves and to ensure we do not violate the general position requirement of Section 2.2. Difficult and subtle issues can arise here, and much work has been done to ameliorate them; see [44] for a survey. We took the approach of having all coordinates in \mathbb{Q} so that so we can do these computations exactly. This entails a stiff speed penalty and leads to points represented by rational numbers with overwhelmingly large denominators. We handle such denominators by rounding coordinates so that the denominator is less than 2^{32} . One can certify at each step by simple local tests that this rounding does not change the isotopy type of the link. However, when the input manifold is hyperbolic, we instead certify correctness of the output diagram after the fact by checking that its exterior is homeomorphic to the manifold in part (a) of the input; this is considerably faster than checking at each step.

3.4 Putting the pieces together

Let \mathcal{T} be a layered filling triangulation with arcs encoding the core curves of the filling and let (P_i) be Pachner moves reducing \mathcal{T} to \mathcal{T}_0 . Our process for producing a barycentric link in \mathcal{T}_0 that is isotopic to the initial L is:

■ **Algorithm 1** `with_arcs[apply_Pachner_moves]($\mathcal{T}, (P_i)$)`.

Start with $\mathcal{T}' := \mathcal{T}$ and loop over the P_1, P_2, \dots, P_n as follows:

1. Apply `with_arcs[P_i]` to \mathcal{T}' to get $P_i\mathcal{T}'$ with arcs representing L . Set $\mathcal{T}' := P_i\mathcal{T}'$.
 2. Loop over the tetrahedra τ in \mathcal{T}' , applying `push` and `straighten` until the arcs stabilize.
-

4 Factoring the 2-to-0 move

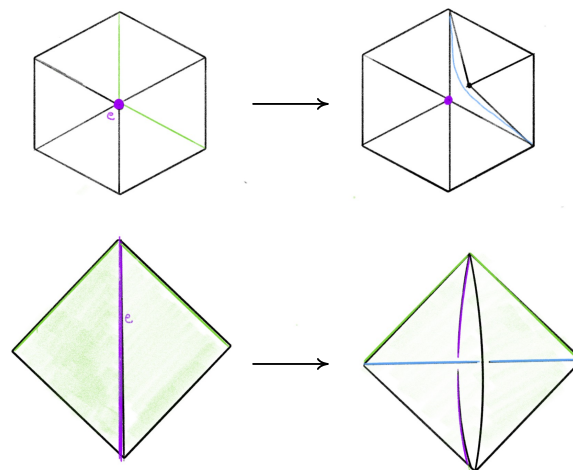
As mentioned in Section 3.1, we factor each $2 \rightarrow 0$ move into a sequence of $2 \rightarrow 3$ and $3 \rightarrow 2$ moves so that we can carry along the barycentric link. This factorization is quite delicate in certain unavoidable corner cases; we outline our method in this section, but leave the details to Appendix C of the full version [18]. To begin to understand the $2 \rightarrow 0$ move, first consider its inverse $0 \rightarrow 2$ move shown in Figure 8. The possible $0 \rightarrow 2$ moves in Figure 8 correspond to a pillow splitting open the *book of tetrahedra* around the edge e . Following [46], we call this pillow a *bird beak* with upper and lower mandibles that pivot around the two outside edges of the beak (viewed from above, these are the purple and black vertices in the top right of Figure 8). On both sides of the bird beak are *half-books* of tetrahedra, together forming a *split-book*. When applying the inverse $2 \rightarrow 0$ move, the two half-books combine to form a book of tetrahedra assembled around the central edge.

The simplest $2 \rightarrow 0$ move is when there are two valence-2 edges that are opposite each other on a single tetrahedron, as shown in Figure 9; equivalently, one of the half-books has a single tetrahedron. This *base case* is handled by Matveev's V move, the composition of four $2 \rightarrow 3$ and $3 \rightarrow 2$ moves of [36, Figure 1.15]. To reduce other instances of the $2 \rightarrow 0$ move to the base case, we rotate a mandible of the bird beak, moving tetrahedra from one half-book to the other until one contains only a single tetrahedron. Because the tetrahedra in the split-book may repeat or be glued together in strange ways, this is rather delicate. When things are sufficiently embedded, Segerman [46] showed:

► **Proposition 4.1.** *Suppose e is a valence-2 edge where the half-books adjacent to the bird beak are embedded and contain m and n tetrahedra respectively. Then the $2 \rightarrow 0$ move can be implemented by $2 \cdot \min(m, n) + 2$ basic $2 \rightarrow 3$ and $3 \rightarrow 2$ moves.*

Proof. We can rotate a mandible by one tetrahedron using the two basic moves of [46, Figure 11]. With $\min(m, n) - 1$ such rotations we can reduce the smaller of the half-books to a single tetrahedron. As already noted, the base case can be done in four moves. ◀

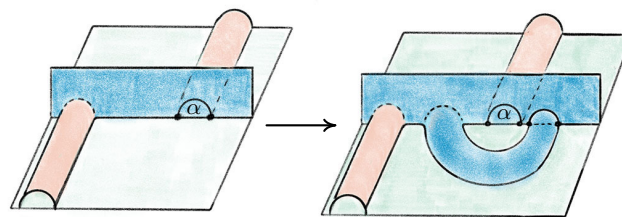
► **Remark 4.2.** One cannot in general factor a $2 \rightarrow 0$ move into a sublinear number of $2 \rightarrow 3$ and $3 \rightarrow 2$ moves: the $2 \rightarrow 0$ move amalgamates two edges of valence $m + 1$ and $n + 1$ into a single edge of valence $m + n$, and each $2 \rightarrow 3$ or $3 \rightarrow 2$ move only changes valences by a total of 12 (counting with multiplicity).



■ **Figure 8** At top, a cross section of a $0 \rightarrow 2$ move; at bottom is a close-up of the inflation of the pillow. The move is performed on the pair of green faces meeting along the purple edge e at left. The resulting pillow is a *bird beak*, which splits open the book of tetrahedra about e . In the top right, the purple and black dots give edges that join together above and below the cross section.

4.1 Twisted beaks and endpoint-through-endpoint moves

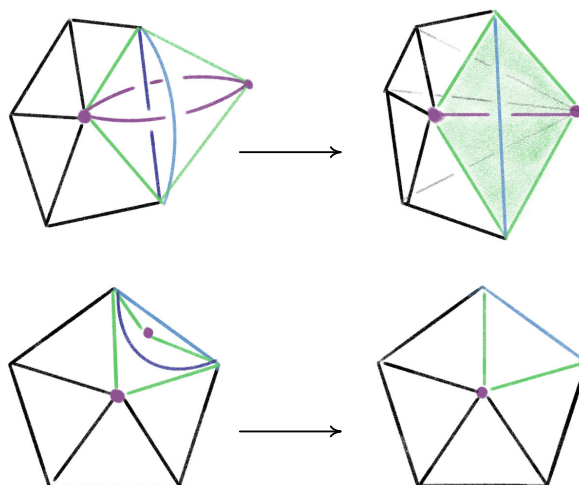
The tricky case is when additional faces of the bird beak are glued to each other. There are two fundamentally different ways for this to happen, shown in Figures 28 and 29 of Appendix C of the full version [18]. The untwisting of these extremely confusing arrangements is done by the endpoint-through-endpoint move of Figure 10, which is in the dual language of special spines from Appendix C of [18]. Matveev’s factorization of the endpoint-through-endpoint move is described in Figure 1.19 of [36]. We simplify this factorization from 14 moves to 6; the key is Proposition C.1 of Appendix C of [18] which shows that



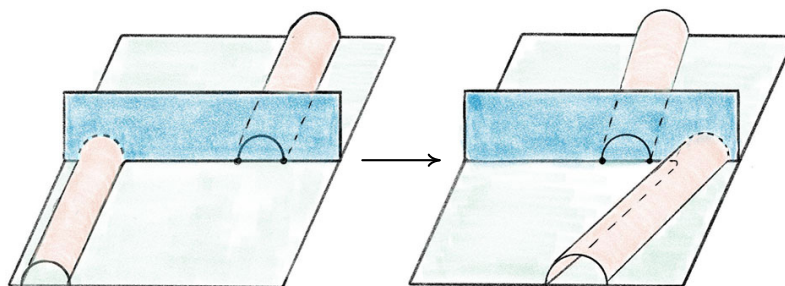
can be done with just two T moves, which is dual to two $2 \rightarrow 3$ Pachner moves. Proposition C.1 in Appendix C of the full version [18] was essential for determining the exact sequence of moves needed to factor the $2 \rightarrow 0$ move. Dual to the endpoint-through-endpoint move are a pair of *untwist the beak* moves, one for each of the situations in Figures 28 and 29, see Appendix C of [18]. We can thus factorize the $2 \rightarrow 0$ move as follows:

5 Building the initial diagram

The base triangulation \mathcal{T}_0 of S^3 has two tetrahedra and one vertex and is shown in Figure 11a; its isomorphism signature in the sense of [9, § 3.2], which completely determines the triangulation, is `cMcabbgdv`. We next give the method for obtaining a planar diagram D for a barycentric link L in \mathcal{T}_0 . We first build a PL link in \mathbb{R}^3 representing L and then project it onto a plane to get D .



■ **Figure 9** The base case of the $2 \rightarrow 0$ move at top with the cross section at bottom.

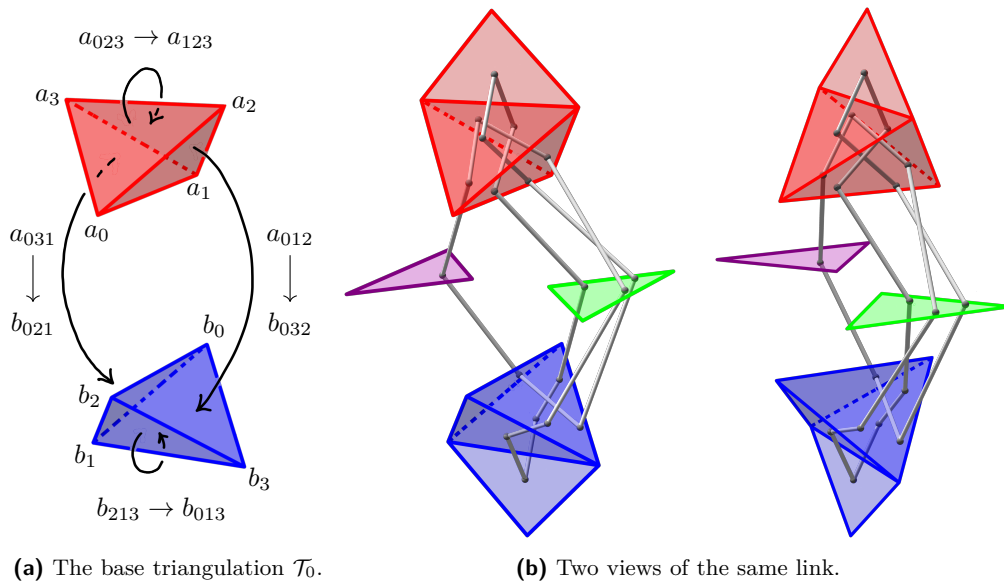


■ **Figure 10** The *endpoint-through-endpoint* move in a special spine.

We cut open \mathcal{T}_0 along its faces and embed the resulting pair of tetrahedra in \mathbb{R}^3 as shown in Figure 11a. This cuts open the link L along its intersections with the faces of \mathcal{T}_0 , resulting in a collection of curves in \mathbb{R}^3 inside the two tetrahedra. To reconnect these curves and recover L , we use *fins* and *lenses* as shown in Figure 11b to interpolate between pairs of faces that are identified in \mathcal{T}_0 . There are two triangular fins, one attached vertically to each tetrahedron, with each fin corresponding to one of the two valence-1 edges of \mathcal{T}_0 . The gluing of two faces incident to a valence-1 edge is realized by folding them onto the corresponding fin. Thus for each barycentric arc that ends in a face corresponding to a fin, we add the line segment joining this endpoint of the arc to the corresponding point in the fin.

■ **Algorithm 2** `factor[2 \rightarrow 0]`.

1. If we are in the base case, do the sequence of moves in the triangulation dual to the factorization of the V move in Figure 1.15 of [36] and exit.
2. If we are in the twisted cases described by Figures 28 and 29 in Appendix C of the full version [18], do the appropriate untwist the beak move. Otherwise, rotate the mandible by one tetrahedron.
3. Go to step 1.



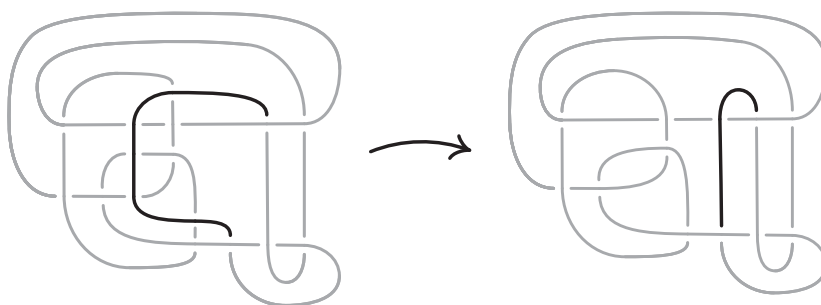
■ **Figure 11** The base triangulation \mathcal{T}_0 in \mathbb{R}^3 , with fins and lenses shown in the middle and at left.

The two triangular lenses lie between the two tetrahedra in a horizontal plane. There is an affine map taking the corresponding face in the top tetrahedron to its lens and a second affine map taking the lens to the corresponding face in the bottom tetrahedron, arranged so their composition is the face pairing in \mathcal{T}_0 . For every arc in the top tetrahedron ending on a face corresponding to a lens, we add the line segment between the endpoint and its image under the affine map to the lens. For each such segment that terminates on a lens, we add the line segment from this endpoint to its image in the face of the bottom tetrahedron under the affine map. This results in a PL link in $\mathbb{R}^3 \subset S^3$ that must be isotopic to L : just imagine puffing out the two tetrahedra to fill all of S^3 following the guides given by the fins and lenses.

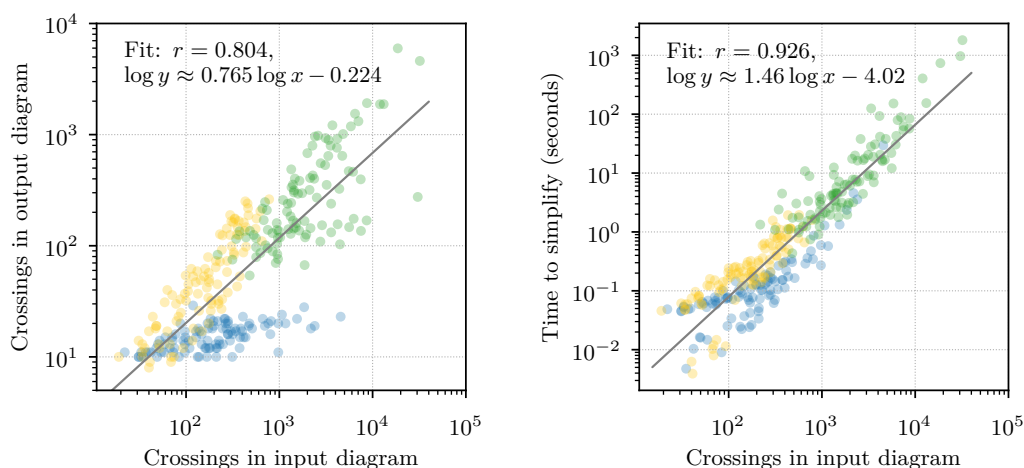
Given a collection of line segments in \mathbb{R}^3 corresponding to the link L , we can build a diagram for L by projecting the line segments onto a plane, computing the crossing information, and assembling this into a planar diagram. Our default choice is roughly to project onto the plane of the page in Figure 11b, with the (so far unused) fall-back of a small random matrix in $SL_3\mathbb{Z}$ if a general-position failure occurs. The link diagrams resulting from this process have many more crossings than is necessary, and we deal with this in Section 6. Still, the specific configuration of fins, lenses, and projection were chosen to try minimize the number of crossings created at this stage; our initial approach used a more compact embedding where the tetrahedra shared a face, and this produced much larger diagrams.

6 Simplifying link diagrams

We now sketch how we simplified the initial link diagram constructed in Section 5, which sometimes had 10,000–100,000 crossings, to produce the final output of our algorithm for FIND DIAGRAM. Previous computational work focused on simplifying diagrams with 20 or fewer crossings [30, 11]. In that regime, random Reidemeister moves combined with flypes are extremely effective in reducing the number of crossings. However, these techniques alone proved inadequate for our much larger links. Instead, we used the more global *strand pickup* method of Figure 12. This technique was introduced by the third author and included in



■ **Figure 12** An example of the strand pickup method for diagram simplification. At left, an *overstrand*, which runs over each crossing it participates in, is indicated by the darker line. At right is the result of isotoping the overstrand, fixing its endpoints, to get a diagram with fewer crossings. The best possible location for an overstrand can be found by solving a weighted shortest-path problem in the planar dual graph to the original diagram.



■ **Figure 13** Simplifying 300 diagrams with between 19 and 32,095 crossings, drawn from Sections 8 and 9.3. The dramatic amount of simplification is shown at left, with an n -crossing knot turned into one with $O(n^{0.8})$ crossings. The running time at right is roughly $O(n^{1.5})$.

SnapPy [15] since version 2.3 (2015), but not previously documented in the literature. It has similarities with the arc representation/grid diagram approach of [20, 21, 22], but it works with arbitrary planar diagrams. When applying the pickup move, we start with the longest overstrands and work towards the shorter ones if no improvement is made. When a pickup move succeeds, we do more basic simplifications before looking for another pickup move. We also do the same move on understrands, going back and forth between the two sides until the diagram stabilizes; for details, see [38]. The high amount of simplification and sub-quadratic running time are shown in Figure 13. As further evidence of its utility, we note that it strictly monotonically reduces the unknot diagrams D_{28} , D_{43} , and PZ_{78} in [12] to the trivial diagram; in contrast, these require adding at least three crossings if one uses only Reidemeister moves.

7 Finding certificates

Part (b) of the input to our algorithm is a certificate that the Dehn filling $M = \hat{M}(\alpha)$ is S^3 in the form of Pachner moves simplifying a triangulation \mathcal{T} of M to the base triangulation \mathcal{T}_0 of S^3 . In practice, one starts with an ideal triangulation $\hat{\mathcal{T}}$ and Dehn filling slopes α where it is unknown if $M(\alpha)$ is S^3 . We therefore need a way of finding this sequence of Pachner moves when it exists. While deciding if a closed 3-manifold M is the S^3 is in **NP** by [31, 45] and additionally in **co-NP** assuming the Generalized Riemann Hypothesis [54, Theorem 11.2], no sub-exponential time algorithm is known. The current best algorithm for S^3 recognition is to heuristically simplify the input triangulation using Pachner moves and then apply the theory of almost normal surfaces, see Algorithm 3.2 of [10]. However, triangulations of S^3 that are truly hard to simplify using Pachner moves have not been encountered in practice, and it is open whether they exist at all [9]. Thus, when M is S^3 , the initial stage of Algorithm 3.2 of [10] nearly always arrives at a 1-tetrahedron triangulation of S^3 and no normal surface theory is needed. The usefulness of our algorithm for FIND DIAGRAM relies on the fact that a heuristic search using Pachner moves gives a practical recognition algorithm for S^3 .

► **Remark 7.1.** The effectiveness of our heuristic search procedure relies on the $2 \rightarrow 0$ move being atomic. Initially, we tried restricting our heuristic search to just the simple Pachner moves, but were typically unable to find a sequence that simplified the input triangulation of S^3 down to one with just a few tetrahedra. (To square this with [9], note from Figure 16 that our triangulations are much larger.) As is clear from Appendix C of the full version [18], factoring the $2 \rightarrow 0$ move as a sequence of $2 \rightarrow 3$ and $3 \rightarrow 2$ moves is complicated enough that one cannot expect to stumble upon these sequences when the triangulation is large and the search is restricted to simple Pachner moves.

Our simplification heuristic closely follows that of SnapPy [15], with some modifications that reduce the complexity of the final barycentric link in \mathcal{T}_0 . These include:

1. Simplifying the layered filling triangulation \mathcal{T} of Section 2.3 as much as possible without modifying the few tetrahedra containing the initial link.
2. Finding sequences of Pachner moves to \mathcal{T}_0 for several different layered filling triangulations, and then using the one requiring the fewest moves for the computations in Sections 3–6.
3. Ensuring the tail of the sequence of moves is a geodesic in the Pachner graph of [9].

The details are in Appendix D of the full version [18].

8 Implementation and initial experiments

We implemented our algorithm in Python, building on the pure-Python `t3mlite` library for 3-manifold triangulations that is part of SnapPy [15]. We also used SnapPy's C kernel to produce the layered filled triangulation \mathcal{T} of Section 2.3 from the input ideal triangulation $\hat{\mathcal{T}}$. The needed linear algebra over \mathbb{Q} was handed by PARI [48]. Not including these libraries, our implementation consists of 1,800 lines of Python code. We had to put some effort into optimization to handle things as large as Figure 3, but more could be done. Our code and data is archived at [19] and the code will be incorporated into version 3.1 of SnapPy [15].

To validate our implementation, we applied it to two samples, one where the input is small and one where the best-possible output is small. The first, \mathcal{CK} , is the 1,267 hyperbolic knots whose exteriors have ideal triangulations with at most 9 tetrahedra [17, 5]. The second, \mathcal{SK} , consists of 1,000 knots whose minimal crossing number was between 10 and 19. There are 100 knots for each crossing number, which were selected at random from all the hyperbolic nonalternating knots with that crossing number [11]; the exception is that there

are only 41 such 10-crossing knots, so 59 alternating 10-crossing knots were used as well. (Alternating knots have unusually close connections between their diagrams and exteriors, so were excluded as possibly being an easy case for FIND DIAGRAM.)

Our program found diagrams for all 2,267 of these exteriors. The running time was under 20 seconds for 96.7% of them, with a max of 2.5 minutes (CPUs were Intel Xeon E5-2690 v3 at 2.6GHz with 4G of memory per core, circa 2014); see Figure 14. The input ideal triangulations $\tilde{\mathcal{T}}$ had between 2 and 44 tetrahedra, and the resulting layered filling triangulation \mathcal{T} had between 13 and 77 tetrahedra (mean of 31.5), typically 60% larger than $\tilde{\mathcal{T}}$; see Figure 15. The sequence of simple Pachner moves used to reduce \mathcal{T} to \mathcal{T}_0 had length between 39 and 761 (mean of 241.0), see Figure 16; this was typically 7.5 times longer than the initial sequence of Pachner moves that included $2 \rightarrow 0$ moves (Figure 17). For the knots in \mathcal{SK} , we compare the size of the output diagram to the minimal crossing number in Figure 18; the output matched the crossing number for 42.1% of these exteriors, and it was within 3 for 87.8%. For \mathcal{CK} , the number of crossings in the output had max 303, mean 65.9, and median 40.

9 Applications

9.1 Congruence links

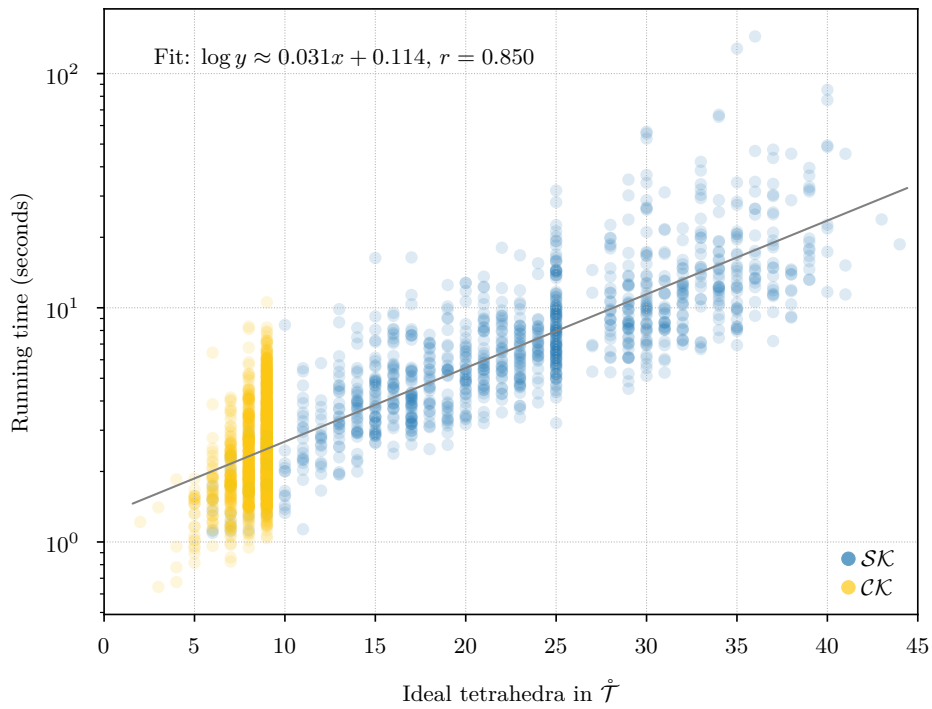
Powerful tools from number theory apply to the special class of arithmetic hyperbolic 3-manifolds. Thurston asked which link exteriors are in the subclass of principal congruence arithmetic manifolds; this was resolved in [6]: there are exactly 48 such exteriors. These 48 have hyperbolic volumes in $[5.33348, 1365.37]$ and ideal triangulations with between 6 and 1,526 tetrahedra. Link diagrams for 15 of these 48 had previously been found by ad hoc methods [7]. Our program has found diagrams for 23 more, including Figures 2 and 3; collectively, we now have links for the 38 such exteriors of smallest volume, see Figure 20.

9.2 Dehn surgery descriptions

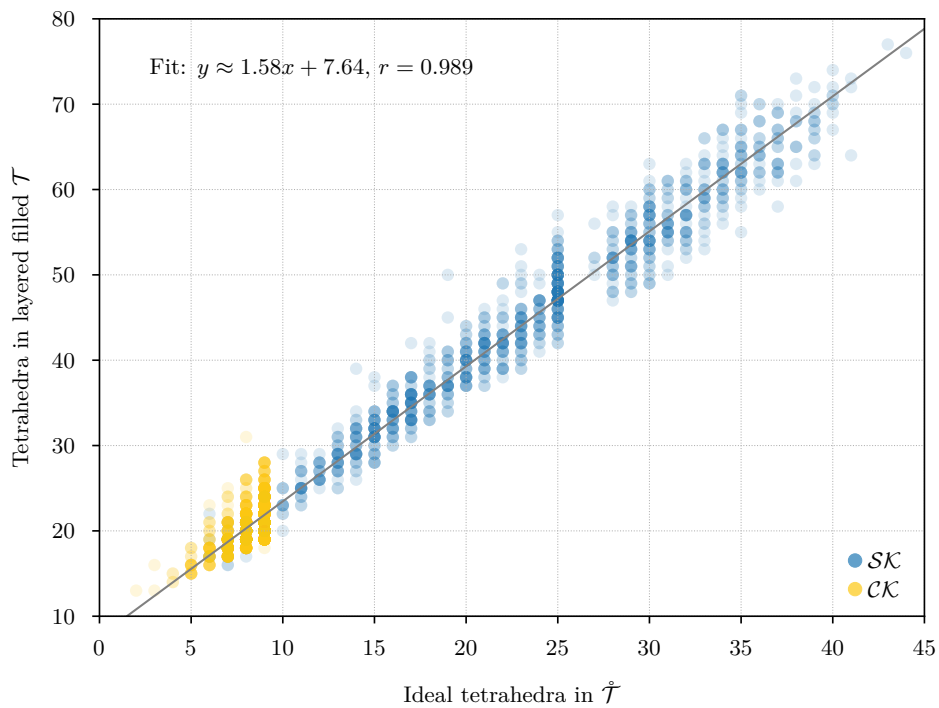
Every closed orientable 3-manifold is Dehn filling on some link exterior in S^3 [43, Chapter 9], and such *Dehn surgery descriptions* play a key role in both theory and practice. However, finding a Dehn surgery description from e.g. a triangulation can be extremely challenging. Thurston observed experimentally that, starting with a closed hyperbolic 3-manifold, one frequently arrives at a link exterior by repeatedly drilling out short closed geodesics, see page 516 of [2]. Combining this with our algorithm for FIND DIAGRAM gives an effective tool for finding Dehn surgery descriptions given a triangulation. We applied this to the Seifert–Weber dodecahedral space, which is an old example [50] still of much current interest [13, 34]. The resulting description in Figure 21 seems to be the first such published; a different description appeared subsequently in [4].

9.3 Knots with the same 0-surgery

The 0-surgery $Z(K)$ on a knot K is the unique Dehn filling N of $E(K)$ where $H_1(N; \mathbb{Q}) \neq 0$. Pairs of knots K and K' with $Z(K)$ homeomorphic to $Z(K')$ are of much interest in low-dimensional topology. Most strikingly, if such a pair K and K' exist with K slice (i.e. bounds a smooth D^2 in D^4) and the Rasmussen s -invariant of K' is nonzero, then the smooth 4-dimensional Poincaré conjecture is false. That is, there would exist a 4-manifold that is homeomorphic but not diffeomorphic to S^4 . See [25, 35] for a general discussion, and also [41] for an important recent result using pairs with $Z(K) \cong Z(K')$. There are many techniques for constructing families of such pairs, which have been unified by the red-blue-green link

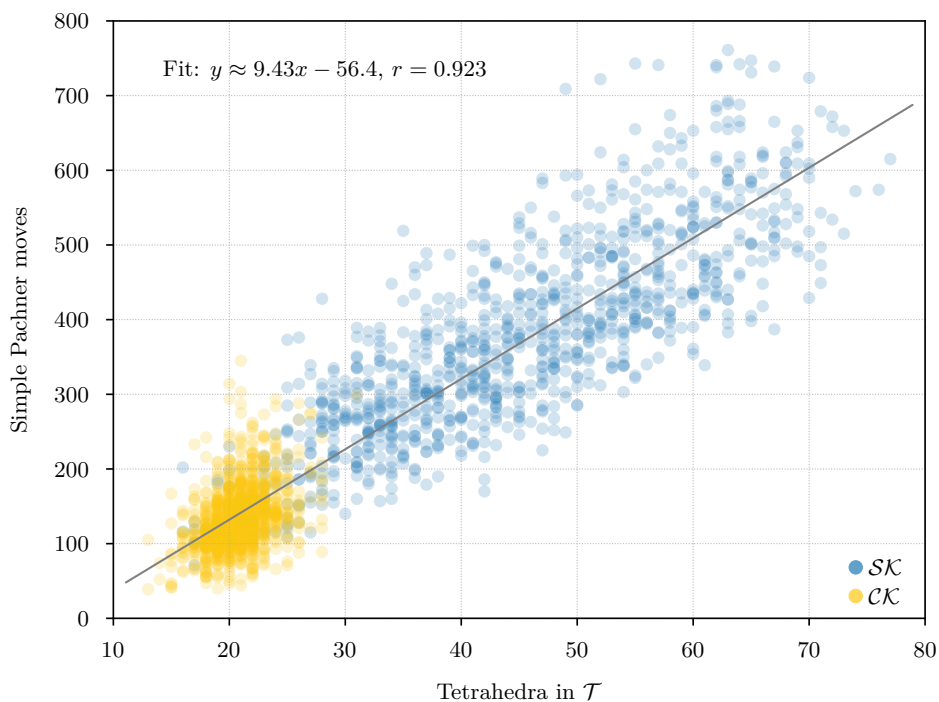


■ **Figure 14** Mean running time for the 2,267 knot exteriors in \mathcal{SK} and \mathcal{CK} appears exponential with small base, roughly $O(1.07^n)$. Compare Figure 19 on the growth of the number of arcs in \mathcal{T}_0 .

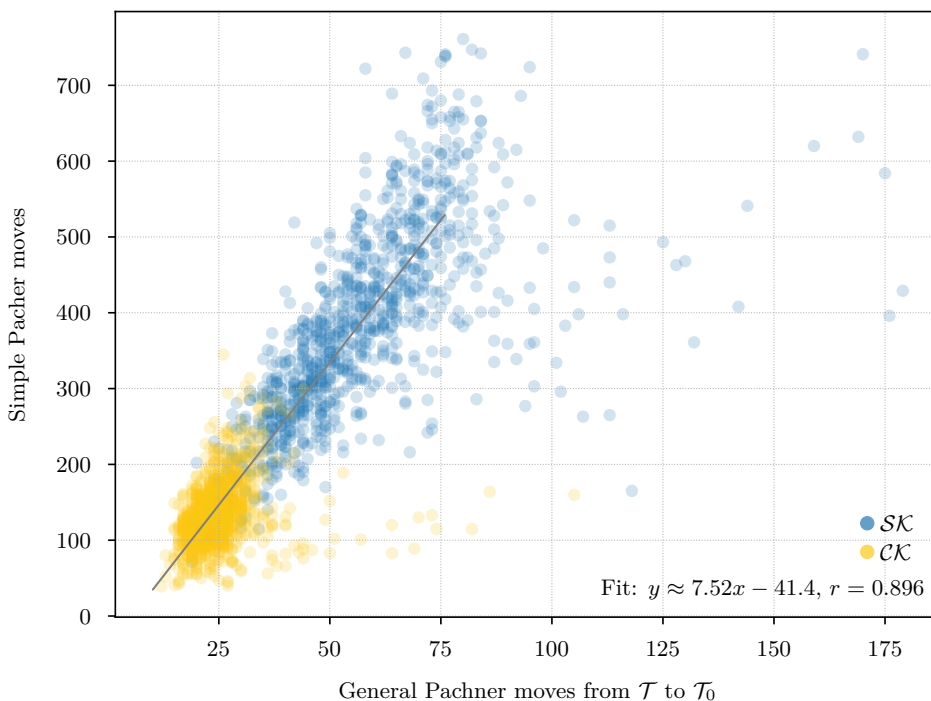


■ **Figure 15** The number of tetrahedra in the layered filled \mathcal{T} compared to the input ideal $\hat{\mathcal{T}}$.

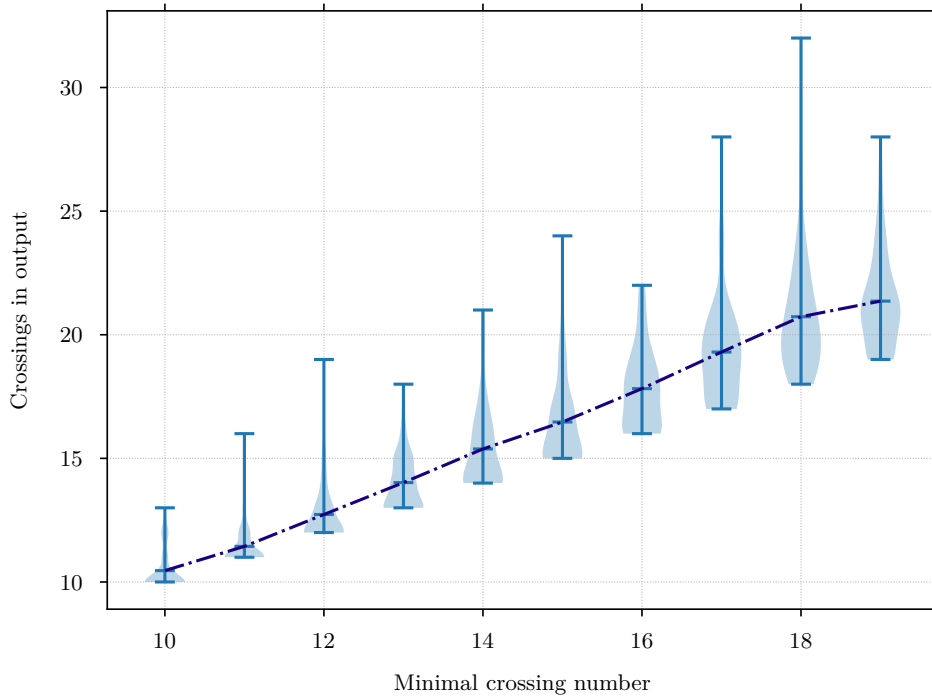
37:18 Computing a Link Diagram from Its Exterior



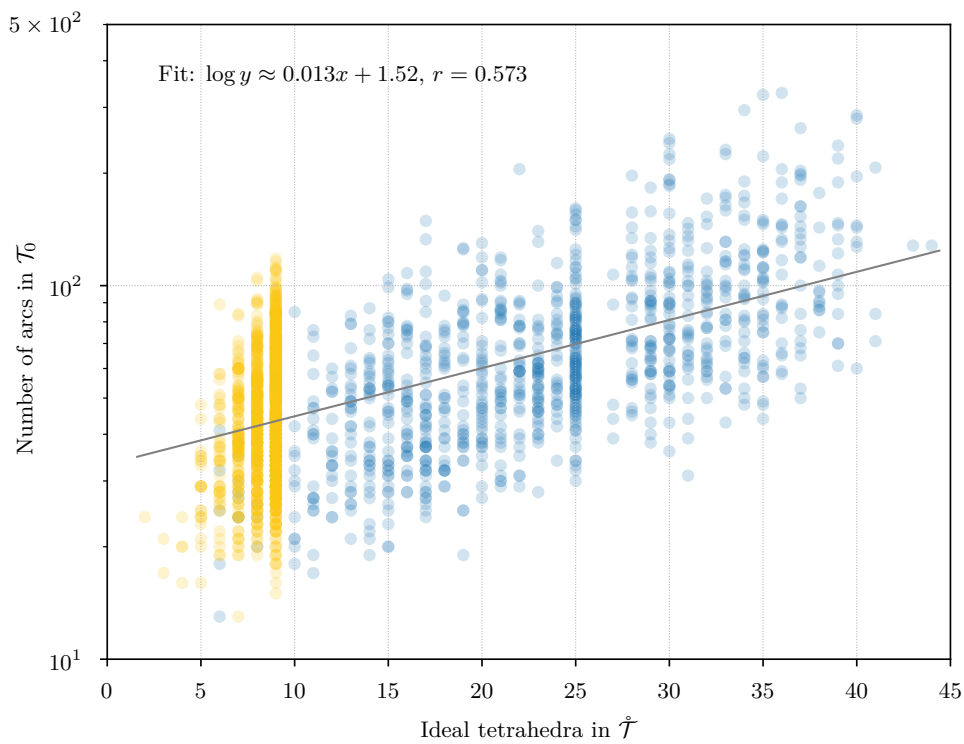
■ **Figure 16** The number of *simple* Pachner moves used to transform the layered filled triangulation \mathcal{T} into the base triangulation \mathcal{T}_0 is generically linear in the size of \mathcal{T} .



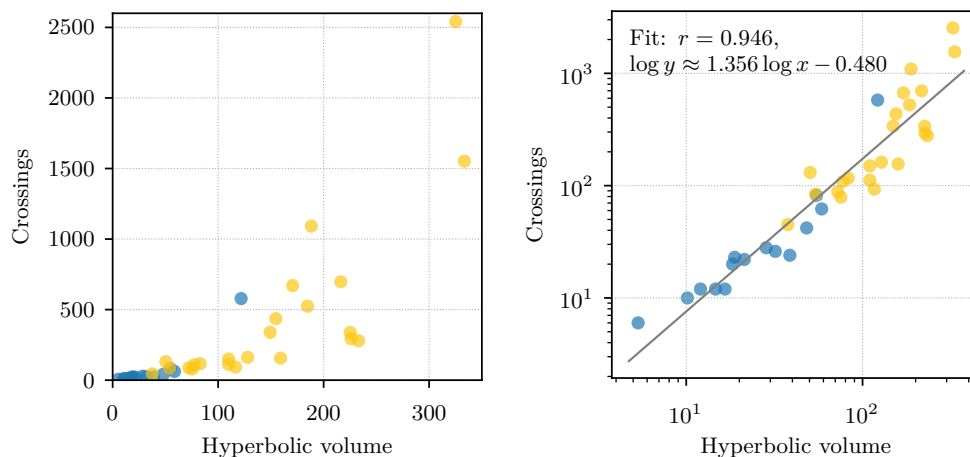
■ **Figure 17** This plot shows the increase in the number of Pachner moves when we factor the $2 \rightarrow 0$ moves into simple Pachner moves. The regression line is based on points with $x < 75$.



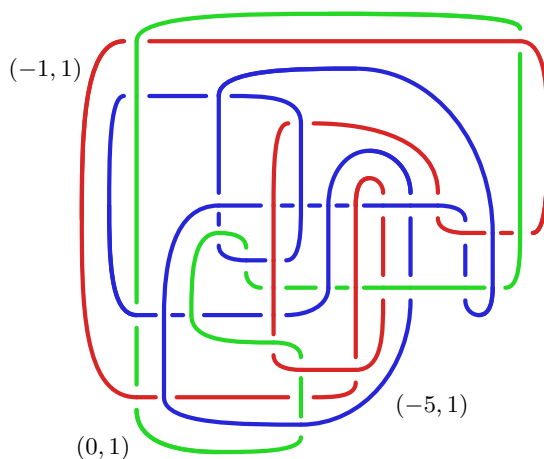
■ **Figure 18** For the knots in SK , grouped by minimum crossing number, the number of crossings in the diagram output by our program. The dotted line indicates the mean.



■ **Figure 19** The number of barycentric arcs when we arrive at \mathcal{T}_0 appears exponential in the size of the input $\hat{\mathcal{T}}$, roughly $O(1.03^n)$.



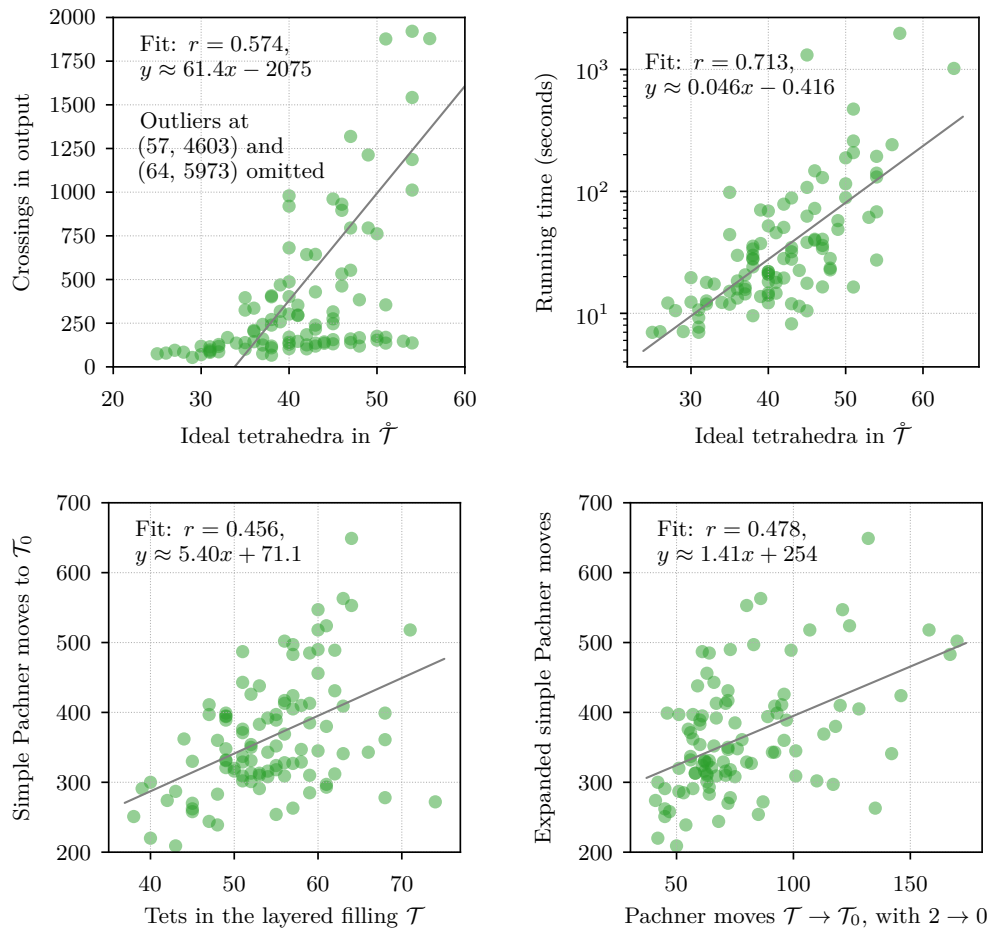
■ **Figure 20** The 38 known link diagrams whose exteriors are principal congruence arithmetic; blue are the 15 from [7], yellow are new. The plots are the same save for the scales on the axes. The regression at right predicts that a link for the largest such exterior would have 9,000 crossings.



■ **Figure 21** A Dehn surgery description of the Seifert–Weber dodecahedral space.

framework of [35]. However, given a particular K , a practical algorithm to search for K' with the same 0-surgery has been lacking. When $Z(K)$ is hyperbolic, we attack this as follows. First, find the short closed geodesics in $Z(K)$ using [29]. Then drill out each geodesic in turn, and test if the resulting manifold \dot{M}' has a Dehn filling which is S^3 ; if it does, use our algorithm for FIND DIAGRAM to \dot{M}' to get a diagram for K' .

Figure 22 shows the result of applying our algorithm to 100 pairs (K, γ) where K is a knot with at most 18 crossings and γ is a short closed geodesic in $Z(K)$ whose exterior is also that of a knot K' in S^3 . In all cases, we were able to recover a diagram for K' , and these were more challenging on average than the examples in Section 8.



■ **Figure 22** Data on the 100 knot exteriors from Section 9.3.

10 Future work

Having demonstrated the practicality of solving FIND DIAGRAM, we plan to refine our implementation and then incorporate it as a standard feature of SnapPy [15] so that it can be widely used. In particular, we aim to:

1. Explore whether the mean running time of $O(1.07^n)$ can be reduced. While Theorem E.1 in Appendix E of the full version [18] shows that the worst case running time must be at least exponential, it is not implausible that the mean running time is polynomial in the size of the *output*. The key issue is that the number of arcs in \mathcal{T}_0 is currently exponential in the size of both the input and the output, compare Figure 19.
2. To reduce the number of arcs, we could consider additional local PL simplification moves, or try the current moves in larger balls in \mathcal{T} made up of several tetrahedra.
3. Explore whether modern methods in computational geometry can be used to speed up the work in Sections 3 and 5.

References

- 1 Colin Adams. Triple crossing number of knots and links. *J. Knot Theory Ramifications*, 22(2):1350006, 17, 2013. doi:10.1142/S0218216513500065.
- 2 Colin C. Adams. Isometric cusps in hyperbolic 3-manifolds. *Michigan Math. J.*, 46(3):515–531, 1999. doi:10.1307/mmj/1030132477.
- 3 Colin Conrad. Adams. *The knot book : an elementary introduction to the mathematical theory of knots*. W.H. Freeman, 1994.
- 4 Kenneth L. Baker. A sketchy surgery description of the seifert-weber dodecahedral space, 2021. URL: <https://sketchsoftopology.wordpress.com/2021/12/09/a-sketchy-surgery>.
- 5 Kenneth L. Baker and Marc Kegel. Census L-space knots are braid positive, except one that is not, in preparation.
- 6 M. D. Baker, M. Goerner, and A. W. Reid. All principal congruence link groups. *J. Algebra*, 528:497–504, 2019. doi:10.1016/j.jalgebra.2019.02.023.
- 7 Mark D. Baker, Matthias Goerner, and Alan W. Reid. All known principal congruence links. Preprint 2019, 9 pages. arXiv:1902.04426.
- 8 Benjamin A. Burton. The cusped hyperbolic census is complete. Preprint 2014, 32 pages. arXiv:1405.2695.
- 9 Benjamin A. Burton. The Pachner graph and the simplification of 3-sphere triangulations. In *Computational geometry (SCG'11)*, pages 153–162. ACM, New York, 2011. doi:10.1145/1998196.1998220.
- 10 Benjamin A. Burton. Computational topology with Regina: algorithms, heuristics and implementations. In *Geometry and topology down under*, volume 597 of *Contemp. Math.*, pages 195–224. Amer. Math. Soc., Providence, RI, 2013. doi:10.1090/conm/597/11877.
- 11 Benjamin A. Burton. The next 350 million knots. In *36th International Symposium on Computational Geometry*, volume 164 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 25, 17. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020. doi:10.4230/LIPIcs.SoCG.2020.25.
- 12 Benjamin A. Burton, Hsien-Chih Chang, Maarten Löffler, Arnaud de Mesmay, Clément Maria, Saul Schleimer, Eric Sedgwick, and Jonathan Spreer. Hard diagrams of the unknot. Preprint 2021, 26 pages. arXiv:2104.14076.
- 13 Benjamin A. Burton, J. Hyam Rubinstein, and Stephan Tillmann. The Weber-Seifert dodecahedral space is non-Haken. *Trans. Amer. Math. Soc.*, 364(2):911–932, 2012. doi:10.1090/S0002-9947-2011-05419-X.
- 14 Abhijit Champanerkar, Ilya Kofman, and Timothy Mullen. The 500 simplest hyperbolic knots. *J. Knot Theory Ramifications*, 23(12):1450055, 34, 2014. doi:10.1142/S0218216514500552.
- 15 Marc Culler, Nathan M. Dunfield, Matthias Goerner, and Jeffrey R. Weeks. SnapPy, a computer program for studying the geometry and topology of 3-manifolds, version 3.0.2, 2021. URL: <https://snappy.computop.org>.
- 16 Arnaud de Mesmay, Yo'av Rieck, Eric Sedgwick, and Martin Tancer. The unbearable hardness of unknotting. *Adv. Math.*, 381:Paper No. 107648, 36, 2021. doi:10.1016/j.aim.2021.107648.
- 17 Nathan M. Dunfield. A census of exceptional Dehn fillings. In *Characters in low-dimensional topology*, volume 760 of *Contemp. Math.*, pages 143–155. Amer. Math. Soc., [Providence], RI, 2020. doi:10.1090/conm/760/15289.
- 18 Nathan M. Dunfield, Malik Obeidin, and Cameron Gates Rudd. Computing a Link Diagram from its Exterior, 2021. Full version of this paper, 34 pages. arXiv:2112.03251v2.
- 19 Nathan M. Dunfield, Malik Obeidin, and Cameron Gates Rudd. Code and data for computing a link diagram from its exterior, 2022. doi:10.7910/DVN/BT1M8R.
- 20 I. A. Dynnikov. Three-page approach to knot theory. Coding and local motions. *Funktsional. Anal. i Prilozhen.*, 33(4):25–37, 96, 1999. doi:10.1007/BF02467109.
- 21 I. A. Dynnikov. Arc-presentations of links: monotonic simplification. *Fund. Math.*, 190:29–76, 2006. doi:10.4064/fm190-0-3.

- 22 Ivan Dynnikov and Vera Sokolova. Multiflypes of rectangular diagrams of links. *J. Knot Theory Ramifications*, 30(6):Paper No. 2150038, 15, 2021. doi:10.1142/S0218216521500383.
- 23 Erica Flapan. *When topology meets chemistry*. Outlooks. Cambridge University Press, Cambridge; Mathematical Association of America, Washington, DC, 2000. A topological look at molecular chirality. doi:10.1017/CB09780511626272.
- 24 Erica Flapan, Adam He, and Helen Wong. Topological descriptions of protein folding. *Proc. Natl. Acad. Sci. USA*, 116(19):9360–9369, 2019. doi:10.1073/pnas.1808312116.
- 25 Michael Freedman, Robert Gompf, Scott Morrison, and Kevin Walker. Man and machine thinking about the smooth 4-dimensional Poincaré conjecture. *Quantum Topol.*, 1(2):171–208, 2010. doi:10.4171/QT/5.
- 26 C. McA. Gordon and J. Luecke. Knots are determined by their complements. *J. Amer. Math. Soc.*, 2(2):371–415, 1989. doi:10.2307/1990979.
- 27 Wolfgang Haken. Theorie der Normalflächen. *Acta Math.*, 105:245–375, 1961. doi:10.1007/BF02559591.
- 28 Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger. The computational complexity of knot and link problems. *J. ACM*, 46(2):185–211, 1999. doi:10.1145/301970.301971.
- 29 Craig D. Hodgson and Jeffrey R. Weeks. Symmetries, isometries and length spectra of closed hyperbolic three-manifolds. *Experiment. Math.*, 3(4):261–274, 1994.
- 30 Jim Hoste, Morwen Thistlethwaite, and Jeff Weeks. The first 1,701,936 knots. *Math. Intelligencer*, 20(4):33–48, 1998. doi:10.1007/BF03025227.
- 31 S. V. Ivanov. The computational complexity of basic decision problems in 3-dimensional topology. *Geom. Dedicata*, 131:1–26, 2008. doi:10.1007/s10711-007-9210-4.
- 32 William Jaco and J. Hyam Rubinstein. Inflations of ideal triangulations. *Adv. Math.*, 267:176–224, 2014. doi:10.1016/j.aim.2014.09.001.
- 33 William Jaco and Eric Sedgwick. Decision problems in the space of Dehn fillings. *Topology*, 42(4):845–906, 2003. doi:10.1016/S0040-9383(02)00083-6.
- 34 Francesco Lin and Michael Lipnowski. Monopole Floer Homology, Eigenform Multiplicities and the Seifert-Weber Dodecahedral Space. *Int. Math. Res. Notices*, to appear. doi:10.1093/imrn/rnaa310.
- 35 Ciprian Manolescu and Lisa Piccirillo. From zero surgeries to candidates for exotic definite four-manifolds. Preprint 2021, 30 pages. arXiv:2102.04391.
- 36 Sergei Matveev. *Algorithmic topology and classification of 3-manifolds*, volume 9 of *Algorithms and Computation in Mathematics*. Springer, Berlin, second edition, 2007.
- 37 Aleksandar Mijatović. Simplifying triangulations of S^3 . *Pacific J. Math.*, 208(2):291–324, 2003. doi:10.2140/pjm.2003.208.291.
- 38 Malik Obeidin. Link simplification code for Spherogram. URL: https://github.com/3-manifolds/Spherogram/blob/master/spherogram_src/links/simplify.py.
- 39 Udo Pachner. P.L. homeomorphic manifolds are equivalent by elementary shellings. *European J. Combin.*, 12(2):129–145, 1991. doi:10.1016/S0195-6698(13)80080-7.
- 40 Satya R. T. Peddada, Nathan M. Dunfield, Lawrence E. Zeidner, Kai A. James, and James T. Allison. Systematic Enumeration and Identification of Unique Spatial Topologies of 3D Systems Using Spatial Graph Representations. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 3A: 47th Design Automation Conference (DAC), 2021. doi:10.1115/DETC2021-66900.
- 41 Lisa Piccirillo. The Conway knot is not slice. *Ann. of Math. (2)*, 191(2):581–591, 2020. doi:10.4007/annals.2020.191.2.5.
- 42 Riccardo Piergallini. Standard moves for standard polyhedra and spines. *Rend. Circ. Mat. Palermo (2) Suppl.*, 18:391–414, 1988. Third National Conference on Topology (Italian) (Trieste, 1986).
- 43 Dale Rolfsen. *Knots and links*, volume 7 of *Mathematics Lecture Series*. Publish or Perish, Inc., Houston, TX, 1990. Corrected reprint of the 1976 original.

- 44 Stefan Schirra. Robustness and precision issues in geometric computation. In *Handbook of computational geometry*, pages 597–632. North-Holland, Amsterdam, 2000. doi:10.1016/B978-044482537-7/50015-2.
- 45 Saul Schleimer. Sphere recognition lies in NP. In *Low-dimensional and symplectic topology*, volume 82 of *Proc. Sympos. Pure Math.*, pages 183–213. Amer. Math. Soc., Providence, RI, 2011. doi:10.1090/pspum/082/2768660.
- 46 Henry Segerman. Connectivity of triangulations without degree one edges under 2-3 and 3-2 moves. *Proc. Amer. Math. Soc.*, 145(12):5391–5404, 2017. doi:10.1090/proc/13485.
- 47 Carl Sundberg and Morwen Thistlethwaite. The rate of growth of the number of prime alternating links and tangles. *Pacific J. Math.*, 182(2):329–358, 1998. doi:10.2140/pjm.1998.182.329.
- 48 The PARI Group, Univ. Bordeaux. *PARI/GP version 2.11.4*, 2020. URL: <http://pari.math.u-bordeaux.fr>.
- 49 Stephan Tillmann. Normal surfaces in topologically finite 3-manifolds. *Enseign. Math. (2)*, 54(3-4):329–380, 2008. arXiv:math/0406271.
- 50 C. Weber and H. Seifert. Die beiden Dodekaederräume. *Math. Z.*, 37(1):237–253, 1933. doi:10.1007/BF01474572.
- 51 Jeff Weeks. Computation of hyperbolic structures in knot theory. In *Handbook of knot theory*, pages 461–480. Elsevier B. V., Amsterdam, 2005. doi:10.1016/B978-044451452-3/50011-3.
- 52 Jeffery R. Weeks. Source code file `close_cusp.c` for SnapPea, version 2.5, circa 1995. URL: https://github.com/3-manifolds/SnapPy/blob/master/kernel/kernel_code/.
- 53 Jeffrey R. Weeks. Convex hulls and isometries of cusped hyperbolic 3-manifolds. *Topology Appl.*, 52(2):127–149, 1993. doi:10.1016/0166-8641(93)90032-9.
- 54 Raphael Zentner. Integer homology 3-spheres admit irreducible representations in $SL(2, \mathbb{C})$. *Duke Math. J.*, 167(9):1643–1712, 2018. doi:10.1215/00127094-2018-0004.

On Comparable Box Dimension

Zdeněk Dvořák ✉

Charles University, Prague, Czech Republic

Daniel Gonçalves ✉

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Abhiruk Lahiri ✉

Charles University, Prague, Czech Republic

Jane Tan ✉

Mathematical Institute, University of Oxford, UK

Torsten Ueckerdt ✉

Karlsruhe Institute of Technology, Germany

Abstract

Two boxes in \mathbb{R}^d are *comparable* if one of them is a subset of a translation of the other one. The *comparable box dimension* of a graph G is the minimum integer d such that G can be represented as a touching graph of comparable axis-aligned boxes in \mathbb{R}^d . We show that proper minor-closed classes have bounded comparable box dimension and explore further properties of this notion.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry; Mathematics of computing \rightarrow Graphs and surfaces

Keywords and phrases geometric graphs, minor-closed graph classes, treewidth fragility

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.38

Related Version *Full Version*: <https://arxiv.org/abs/2203.07686>

Funding *Zdeněk Dvořák*: Supported by the ERC-CZ project LL2005 (Algorithms and complexity within and beyond bounded expansion) of the Ministry of Education of Czech Republic.

Daniel Gonçalves: Supported by the ANR grant GATO ANR-16-CE40-0009.

Abhiruk Lahiri: Supported by the ERC-CZ project LL2005 (Algorithms and complexity within and beyond bounded expansion) of the Ministry of Education of Czech Republic.

Acknowledgements This research was carried out at the workshop on Geometric Graphs and Hypergraphs organized by Yelena Yuditsky and Torsten Ueckerdt in September 2021. We would like to thank the organizers and all participants for creating a friendly and productive environment.

1 Introduction

Given a system \mathcal{O} of subsets of \mathbb{R}^d , we say that a graph G is a *touching graph of objects from \mathcal{O}* if there exists a function $f : V(G) \rightarrow \mathcal{O}$ (called a *touching representation by objects from \mathcal{O}*) such that the interiors of $f(u)$ and $f(v)$ are disjoint for all distinct $u, v \in V(G)$, and $f(u) \cap f(v) \neq \emptyset$ if and only if $uv \in E(G)$. Famously, Koebe [13] proved that a graph is planar if and only if it is a touching graph of balls in \mathbb{R}^2 . This result has motivated numerous strengthenings and variations (see [14, 19] for some classical examples); most relevantly for us, Felsner and Francis [11] showed that every planar graph is a touching graph of cubes in \mathbb{R}^3 .

An attractive feature of touching representations is that it is possible to represent graph classes that are sparse (e.g., planar graphs, or more generally, graph classes with bounded expansion [15]). This is in contrast to general intersection representations where the represented class always includes arbitrarily large cliques. Of course, whether the class



© Zdeněk Dvořák, Daniel Gonçalves, Abhiruk Lahiri, Jane Tan, and Torsten Ueckerdt; licensed under Creative Commons License CC-BY 4.0

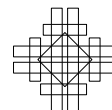
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 38; pp. 38:1–38:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



of touching graphs of objects from \mathcal{O} is sparse or not depends on the particular system \mathcal{O} . For example, all complete bipartite graphs $K_{n,m}$ are touching graphs of boxes in \mathbb{R}^3 , where the vertices in one part are represented by $m \times 1 \times 1$ boxes and the vertices of the other part are represented by $1 \times n \times 1$ boxes (throughout the paper, by *box* we always mean *axis-aligned box*, i.e., the Cartesian product of closed intervals of non-zero length). Dvořák, McCarty and Norin [6] noticed that this issue disappears if we forbid such a combination of long and wide boxes. This condition can be expressed as follows. For two boxes B_1 and B_2 , we write $B_1 \sqsubseteq B_2$ if B_2 contains a translate of B_1 . We say that B_1 and B_2 are *comparable* if $B_1 \sqsubseteq B_2$ or $B_2 \sqsubseteq B_1$. A *touching representation by comparable boxes* of a graph G is a touching representation f by boxes such that for every $u, v \in V(G)$, the boxes $f(u)$ and $f(v)$ are comparable. Let the *comparable box dimension* $\dim_{cb}(G)$ of a graph G be the smallest integer d such that G has a touching representation by comparable boxes in \mathbb{R}^d . We remark that the comparable box dimension of every graph G is at most $|V(G)|$, see Section 3.1 for details. Then, for a class \mathcal{G} of graphs, let $\dim_{cb}(\mathcal{G}) := \sup\{\dim_{cb}(G) : G \in \mathcal{G}\}$. If the comparable box dimension of graphs in \mathcal{G} is not bounded, we write $\dim_{cb}(\mathcal{G}) = \infty$.

Dvořák, McCarty and Norin [6] proved some basic properties of this notion. In particular, they showed that if a class \mathcal{G} has finite comparable box dimension, then it has polynomial strong coloring numbers, which implies that \mathcal{G} has strongly sublinear separators. They also provided an example showing that, for many functions h , the class of graphs with strong coloring numbers bounded by h has infinite comparable box dimension¹. Dvořák et al. [9] proved that graphs of comparable box dimension 3 have exponential weak coloring numbers, giving the first natural graph class with polynomial strong coloring numbers and superpolynomial weak coloring numbers (the previous example is obtained by subdividing edges of every graph suitably many times [12]).

We show that the comparable box dimension behaves well under the operations of addition of apex vertices, clique-sums, and taking subgraphs. Together with known results on product structure [4], this implies the main result of this paper.

► **Theorem 1.** *The comparable box dimension of every proper minor-closed class of graphs is finite.*

Additionally, we show that classes of graphs with finite comparable box dimension are fractionally treewidth-fragile. This gives arbitrarily precise approximation algorithms for all monotone maximization problems that are expressible in terms of distances between the solution vertices and tractable on graphs of bounded treewidth [8], or expressible in the first-order logic [7].

2 Parameters

In this section we bound some basic graph parameters in terms of comparable box dimension. The first result bounds the clique number $\omega(G)$ in terms of $\dim_{cb}(G)$.

► **Lemma 2.** *For any graph G , we have $\omega(G) \leq 2^{\dim_{cb}(G)}$.*

Proof. We may assume that G has bounded comparable box dimension witnessed by a box representation f . To represent any clique $A = \{a_1, \dots, a_w\}$ in G , the corresponding boxes $f(a_1), \dots, f(a_w)$ have pairwise non-empty intersections. Since axis-aligned boxes have the Helly property, there is a point $p \in \mathbb{R}^d$ contained in $f(a_1) \cap \dots \cap f(a_w)$. As each box is

¹ In their construction $h(r)$ has to be at least 3, and has to tend to $+\infty$.

full-dimensional, their interiors each intersect at least one of the 2^d orthants at p . At the same time, it follows from the definition of a touching representation that $f(a_1), \dots, f(a_d)$ have pairwise disjoint interiors, and hence $w \leq 2^d$. ◀

Note that a clique with 2^d vertices has a touching representation by comparable boxes in \mathbb{R}^d , where each vertex is a hypercube defined as the Cartesian product of intervals of form $[-1, 0]$ or $[0, 1]$. From this together with Lemma 2, it follows that $\dim_{cb}(K_{2^d}) = d$.

The remaining bounds pertain to the chromatic number $\chi(G)$ of a graph G , and two of its variants. An *acyclic coloring* (resp. *star coloring*) of a graph G is a proper coloring such that any two color classes induce a forest (resp. star forest, i.e., a forest in which each component is a star). The *acyclic chromatic number* $\chi_a(G)$ (resp. *star chromatic number* $\chi_s(G)$) of G is the minimum number of colors in an acyclic (resp. star) coloring of G . We will need the fact that all the variants of the chromatic number are at most exponential in the comparable box dimension; this follows from [6], although we include an argument to make the dependence clear.

► **Lemma 3.** *For any graph G we have $\chi(G) \leq 3^{\dim_{cb}(G)}$, $\chi_a(G) \leq 5^{\dim_{cb}(G)}$ and $\chi_s(G) \leq 2 \cdot 9^{\dim_{cb}(G)}$.*

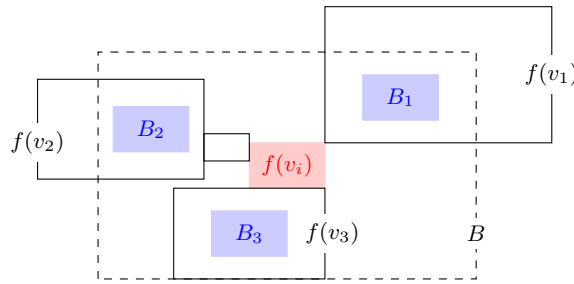
Proof. We focus on the star chromatic number and note that the chromatic number and the acyclic chromatic number may be bounded similarly. Suppose that G has comparable box dimension d witnessed by a representation f , and let v_1, \dots, v_n be the vertices of G written so that $\text{vol}(f(v_1)) \geq \dots \geq \text{vol}(f(v_n))$. Equivalently, we have $f(v_i) \sqsubseteq f(v_j)$ whenever $i > j$. Now define a greedy coloring c so that $c(v_i)$ is the smallest color such that $c(v_i) \neq c(v_j)$ for any $j < i$ for which either $v_j v_i \in E(G)$ or there exists $m > j$ such that $v_j v_m, v_m v_i \in E(G)$. Note that this gives a star coloring, since a path on four vertices always contains a 3-vertex subpath of the form $v_{i_1} v_{i_2} v_{i_3}$ such that $i_1 < i_2, i_3$, and our coloring procedure gives distinct colors to vertices forming such a path.

It remains to bound the number of colors used. Suppose we are coloring v_i . We shall bound the number of vertices v_j such that $j < i$ and such that there exists $m > i$ for which $v_j v_m, v_m v_i \in E(G)$. Let B be the box obtained by scaling up $f(v_i)$ by a factor of 5 while keeping the same center. Since $f(v_m) \sqsubseteq f(v_i) \sqsubseteq f(v_j)$, there exists a translation B_j of $f(v_i)$ contained in $f(v_j) \cap B$ (see Figure 1). Two boxes B_j and $B_{j'}$ for $j \neq j'$ have disjoint interiors since their intersection is contained in the intersection of the touching boxes $f(v_j)$ and $f(v_{j'})$, and their interiors are also disjoint from $f(v_i) \subset B$. Thus, the number of such indices j is at most $\text{vol}(B) / \text{vol}(f(v_i)) - 1 = 5^d - 1$.

A similar argument shows that the number of indices m such that $m < i$ and $v_m v_i \in E(G)$ is at most $3^d - 1$. Consequently, the number of indices $j < i$ for which there exists m such that $j < m < i$ and $v_j v_m, v_m v_i \in E(G)$ is at most $(3^d - 1)^2$. This means that when choosing the color of v_i greedily, we only need to avoid colors of at most $(5^d - 1) + (3^d - 1) + (3^d - 1)^2$ vertices, so $2 \cdot 9^d$ colors suffice. ◀

3 Operations

It is clear that, given a touching representation of a graph G , one can easily obtain a touching representation by boxes of an induced subgraph H of G by simply deleting the boxes corresponding to the vertices in $V(G) \setminus V(H)$. We shall show that these representations also behave nicely under several other basic operations on graphs. To describe the boxes, we shall use the Cartesian product \times defined among boxes of lower dimension (so that $A \times B$



■ **Figure 1** Nearby boxes obstructing colors at v_i .

is the box whose projection on some first number of dimensions gives the box A , while the projection on the remaining dimensions gives the box B , or specify its projections onto every dimension (and in this case write $A[i]$ for the interval obtained from projecting A on its i^{th} dimension).

3.1 Vertex addition

Let us start with a simple lemma which says that the addition of a vertex increases the comparable box dimension by at most one. In particular, this implies that $\text{dim}_{cb}(G) \leq |V(G)|$.

► **Lemma 4.** *For any graph G and $v \in V(G)$, we have $\text{dim}_{cb}(G) \leq \text{dim}_{cb}(G - v) + 1$.*

Proof. Let f be a touching representation of $G - v$ by comparable boxes in \mathbb{R}^d , where $d = \text{dim}_{cb}(G - v)$. We define a representation h of G as follows. For each $u \in V(G) \setminus \{v\}$, let $h(u) = [0, 1] \times f(u)$ if $uv \in E(G)$ and $h(u) = [1/2, 3/2] \times f(u)$ if $uv \notin E(G)$. Let $h(v) = [-1, 0] \times [-M, M] \times \cdots \times [-M, M]$, where M is chosen large enough so that $f(u) \subseteq [-M, M] \times \cdots \times [-M, M]$ for every $u \in V(G) \setminus \{v\}$. Then h is a touching representation of G by comparable boxes in \mathbb{R}^{d+1} . ◀

3.2 Strong product

Let $G \boxtimes H$ denote the *strong product* of the graphs G and H , i.e., the graph with vertex set $V(G) \times V(H)$ and with distinct vertices (u_1, v_1) and (u_2, v_2) adjacent if and only if u_1 is equal to or adjacent to u_2 in G and v_1 is equal to or adjacent to v_2 in H . To obtain a touching representation of $G \boxtimes H$ it suffices to take a product of representations of G and H , but the resulting representation may contain incomparable boxes. Indeed, in general $\text{dim}_{cb}(G \boxtimes H)$ is not bounded by a function of $\text{dim}_{cb}(G)$ and $\text{dim}_{cb}(H)$; for example, every star has comparable box dimension at most two, but the strong product of the star $K_{1,n}$ with itself contains $K_{n,n}$ as an induced subgraph, and thus its comparable box dimension is at least $\Omega(\log n)$. However, as shown in the following lemma, this issue does not arise if the representation of H consists of translates of a single box; by scaling, we can without loss of generality assume this box is a unit hypercube.

► **Lemma 5.** *Consider a graph H having a touching representation h in \mathbb{R}^{d_H} by axis-aligned hypercubes of unit size. Then for any graph G , the strong product $G \boxtimes H$ of these graphs has comparable box dimension at most $\text{dim}_{cb}(G) + d_H$.*

Proof. It suffices to take a product of the two representations. Indeed, consider a touching representation g of G by comparable boxes in \mathbb{R}^{d_G} , with $d_G = \dim_{cb}(G)$, and the representation h of H . Let us define a representation f of $G \boxtimes H$ in $\mathbb{R}^{d_G+d_H}$ by

$$f((u, v))[i] = \begin{cases} g(u)[i] & \text{if } i \leq d_G \\ h(v)[i - d_G] & \text{if } i > d_G. \end{cases}$$

Consider distinct vertices (u, v) and (u', v') of $G \boxtimes H$. The boxes $g(u)$ and $g(u')$ are comparable, say $g(u) \sqsubseteq g(u')$. Since $h(v')$ is a translation of $h(v)$, this implies that $f((u, v)) \sqsubseteq f((u', v'))$. Hence, the boxes of the representation f are pairwise comparable.

The boxes of the representations g and h have pairwise disjoint interiors. Hence, if $u \neq u'$, then there exists $i \leq d_G$ such that the interiors of the intervals $f((u, v))[i] = g(u)[i]$ and $f((u', v'))[i] = g(u')[i]$ are disjoint; if $v \neq v'$, then there exists $i \leq d_H$ such that the interiors of the intervals $f((u, v))[i + d_G] = h(v)[i]$ and $f((u', v'))[i + d_G] = h(v')[i]$ are disjoint. Consequently, the interiors of boxes $f((u, v))$ and $f((u', v'))$ are pairwise disjoint. Moreover, if $u \neq u'$ and $uu' \notin E(G)$, or if $v \neq v'$ and $vv' \notin E(G)$, then the aforementioned intervals (not just their interiors) are disjoint for some i ; hence, if (u, v) and (u', v') are not adjacent in $G \boxtimes H$, then $f((u, v)) \cap f((u', v')) = \emptyset$. Therefore, f is a touching representation of a subgraph of $G \boxtimes H$.

Finally, suppose that (u, v) and (u', v') are adjacent in $G \boxtimes H$. Then there exists a point p_G in the intersection of $g(u)$ and $g(u')$, since $u = u'$ or $uu' \in E(G)$ and g is a touching representation of G ; and similarly, there exists a point p_H in the intersection of $h(v)$ and $h(v')$. Then $p_G \times p_H$ is a point in the intersection of $f((u, v))$ and $f((u', v'))$. Hence, f is indeed a touching representation of $G \boxtimes H$. ◀

3.3 Taking a subgraph

The comparable box dimension of a subgraph of a graph G may be larger than $\dim_{cb}(G)$ (see the end of this section for an example). However, we show that the comparable box dimension of a subgraph is at most exponential in the comparable box dimension of the whole graph. This is essentially Corollary 25 in [6], but since the setting is somewhat different and the construction of [6] uses rotated boxes, we provide details of the argument.

► **Lemma 6.** *If G is a subgraph of a graph G' , then $\dim_{cb}(G) \leq \dim_{cb}(G') + \frac{1}{2}\chi_s^2(G')$.*

Proof. By removing boxes that represent vertices of G that are not in G' , we may assume that $V(G') = V(G)$. Let f be a touching representation of G' by comparable boxes in \mathbb{R}^d , where $d = \dim_{cb}(G')$. Let φ be a star coloring of G' using colors $\{1, \dots, c\}$, where $c = \chi_s(G')$.

For any distinct colors $i, j \in \{1, \dots, c\}$, let $A_{i,j} \subseteq V(G)$ be the set of vertices u of color i such that there exists a vertex v of color j such that $uv \in E(G') \setminus E(G)$. For each $u \in A_{i,j}$, let $a_j(u)$ denote such a vertex v chosen arbitrarily.

Let us define a representation h by boxes in $\mathbb{R}^{d+\binom{c}{2}}$ by starting from the representation f and, for each pair $i < j$ of colors, adding a dimension $d_{i,j}$ and setting

$$h(v)[d_{i,j}] = \begin{cases} [1/3, 4/3] & \text{if } v \in A_{i,j} \\ [-4/3, -1/3] & \text{if } v \in A_{j,i} \\ [-1/2, 1/2] & \text{otherwise.} \end{cases}$$

Note that the boxes in this extended representation are comparable, as in the added dimensions, all the boxes have size 1.

38:6 On Comparable Box Dimension

Suppose $uv \in E(G)$, where $\varphi(u) = i$ and $\varphi(v) = j$ and say $i < j$. We cannot have $u \in A_{i,j}$ and $v \in A_{j,i}$, as then $a_j(u)u v a_i(v)$ would be a 4-vertex path in G' in colors i and j . Hence, in any added dimension d' , we have $h(u)[d'] = [-1/2, 1/2]$ or $h(v)[d'] = [-1/2, 1/2]$, and thus $h(u)[d'] \cap h(v)[d'] \neq \emptyset$. Since the boxes $f(u)$ and $f(v)$ touch, it follows that the boxes $h(u)$ and $h(v)$ touch as well.

Suppose now that $uv \notin E(G)$. If $uv \notin E(G')$, then $f(u)$ is disjoint from $f(v)$, and thus $h(u)$ is disjoint from $h(v)$. Hence, we can assume $uv \in E(G') \setminus E(G)$, $\varphi(u) = i$, $\varphi(v) = j$ and $i < j$. Then $u \in A_{i,j}$, $v \in A_{j,i}$, $h(u)[d_{i,j}] = [1/3, 4/3]$, $h(v)[d_{j,i}] = [-4/3, -1/3]$, and $h(u) \cap h(v) = \emptyset$.

Consequently, h is a touching representation of G by comparable boxes in dimension $d + \binom{c}{2} \leq d + c^2/2$. ◀

Let us now combine Lemmas 3 and 6.

► **Corollary 7.** *If G is a subgraph of a graph G' , then $\dim_{cb}(G) \leq \dim_{cb}(G') + 2 \cdot 81^{\dim_{cb}(G')} \leq 3 \cdot 81^{\dim_{cb}(G')}$.*

An exponential increase in the dimension is unavoidable: we have $\dim_{cb}(K_{2^d}) = d$, but the graph obtained from K_{2^d} by deleting a perfect matching has comparable box dimension 2^{d-1} . Indeed, for every pair u, v of non-adjacent vertices there is a specific dimension i such that their boxes span intervals $[a, b]$ and $[c, d]$ with $b < c$, while the i^{th} interval of every other box in the representation contains $[b, c]$.

3.4 Clique-sums

A *clique-sum* of two graphs G_1 and G_2 is obtained from their disjoint union by identifying vertices of a clique in G_1 and a clique of the same size in G_2 and possibly deleting some of the edges of the resulting clique. A *full clique-sum* is a clique-sum in which we keep all the edges of the resulting clique. The main issue to overcome in obtaining a representation for a (full) clique-sum is that the representations of G_1 and G_2 can be “degenerate”. Consider, for example, the case where G_1 is represented by unit squares arranged in a grid; here there is no space to attach G_2 at the cliques formed by four squares intersecting in a single corner. This can be avoided by increasing the dimension, but we need to be careful so that the dimension stays bounded even after an arbitrary number of clique-sums. We thus introduce the notion of *clique-sum extendable* representations.

► **Definition 8.** *Consider a graph G with a distinguished clique C^* , called the root clique of G . A touching representation h of G by (not necessarily comparable) boxes in \mathbb{R}^d is called C^* -clique-sum extendable if the following conditions hold for every sufficiently small $\varepsilon > 0$.*

(vertices) *For each $u \in V(C^*)$, there exists a dimension d_u , such that:*

(v0) $d_u \neq d_{u'}$ for distinct $u, u' \in V(C^*)$,

(v1) each vertex $u \in V(C^*)$ satisfies $h(u)[d_u] = [-1, 0]$ and $h(u)[i] = [0, 1]$ for any dimension $i \neq d_u$, and

(v2) each vertex $v \notin V(C^*)$ satisfies $h(v) \subset [0, 1]^d$.

(cliques) *For every clique C of G , there exists a point $p(C) \in [0, 1]^d \cap \left(\bigcap_{v \in V(C)} h(v)\right)$ such that, defining the clique box $h^\varepsilon(C)$ by setting $h^\varepsilon(C)[i] = [p(C)[i], p(C)[i] + \varepsilon]$ for every dimension i , the following conditions are satisfied:*

(c1) *For any two cliques $C_1 \neq C_2$, $h^\varepsilon(C_1) \cap h^\varepsilon(C_2) = \emptyset$ (equivalently, $p(C_1) \neq p(C_2)$).*

(c2) A box $h(v)$ intersects $h^\varepsilon(C)$ if and only if $v \in V(C)$, and in that case their intersection is a facet of $h^\varepsilon(C)$ incident to $p(C)$. That is, there exists a dimension $i_{C,v}$ such that for each dimension j ,

$$h(v)[j] \cap h^\varepsilon(C)[j] = \begin{cases} \{p(C)[i_{C,v}]\} & \text{if } j = i_{C,v} \\ [p(C)[j], p(C)[j] + \varepsilon] & \text{otherwise.} \end{cases}$$

Note that the root clique can be empty, that is the empty subgraph with no vertices. In that case the clique is denoted \emptyset . Let $\dim_{cb}^{ext}(G)$ be the minimum dimension such that G has an \emptyset -clique-sum extendable touching representation by comparable boxes.

Let us remark that a clique-sum extendable representation in dimension d implies the existence of such a representation in higher dimensions as well.

► **Lemma 9.** *Let G be a graph with a root clique C^* and let h be a C^* -clique-sum extendable touching representation of G by comparable boxes in \mathbb{R}^d . Then G has such a representation in $\mathbb{R}^{d'}$ for every $d' \geq d$.*

Proof. It clearly suffices to consider the case that $d' = d + 1$. Note that the (vertices) conditions imply that $h(v') \sqsubseteq h(v)$ for every $v' \in V(G) \setminus V(C^*)$ and $v \in V(C^*)$. We extend the representation h by setting $h(v)[d + 1] = [0, 1]$ for $v \in V(C^*)$ and $h(v)[d + 1] = [0, \frac{1}{2}]$ for $v \in V(G) \setminus V(C^*)$. The clique point $p(C)$ of each clique C is extended by setting $p(C)[d + 1] = \frac{1}{4}$. It is easy to verify that the resulting representation is C^* -clique-sum extendable. ◀

The following lemma ensures that clique-sum extendable representations behave well with respect to full clique-sums. The proof is omitted, but the key strategy is to translate (allowing exchanges of dimensions) and scale h_2 to fit in $h_1^\varepsilon(C_1)$.

► **Lemma 10.** *Consider two graphs G_1 and G_2 , given with a C_1^* - and a C_2^* -clique-sum extendable representations h_1 and h_2 by comparable boxes in \mathbb{R}^{d_1} and \mathbb{R}^{d_2} , respectively. Let G be the graph obtained by performing a full clique-sum of these two graphs on any clique C_1 of G_1 , and on the root clique C_2^* of G_2 . Then G admits a C_1^* -clique sum extendable representation h by comparable boxes in $\mathbb{R}^{\max(d_1, d_2)}$.*

Moreover, we can pick the root clique at the expense of increasing the dimension by $\omega(G)$. This proof is also omitted, but it is essentially the same as that of Lemma 4.

► **Lemma 11.** *For any graph G and any clique C^* , the graph G admits a C^* -clique-sum extendable touching representation by comparable boxes in \mathbb{R}^d , for $d = |V(C^*)| + \dim_{cb}^{ext}(G \setminus V(C^*))$.*

The last key lemma that we will need in this section is an upper bound on $\dim_{cb}^{ext}(G)$ in terms of $\dim_{cb}(G)$ and $\chi(G)$.

► **Lemma 12.** *For any graph G , $\dim_{cb}^{ext}(G) \leq \dim_{cb}(G) + \chi(G)$.*

Proof. Let h be a touching representation of G by comparable boxes in \mathbb{R}^d , with $d = \dim_{cb}(G)$, and let c be a $\chi(G)$ -coloring of G . We start with a slightly modified version of h . We first scale h to fit in $(0, 1)^d$, and for a sufficiently small real $\alpha > 0$ we increase each box in h by 2α in every dimension, that is we replace $h(v)[i] = [a, b]$ by $[a - \alpha, b + \alpha]$ for each vertex v and dimension i . Here, we choose α to be sufficiently small so that the boxes representing non-adjacent vertices remain disjoint, and thus the resulting representation h_1 is an intersection representation of the same graph G . Moreover, observe that for every clique

38:8 On Comparable Box Dimension

C of G , the intersection $I_C = \bigcap_{v \in V(C)} h_1(v)$ is a box with non-zero edge lengths. For any clique C of G , let $p_1(C)$ be a point in the interior of I_C different from the points chosen for all other cliques.

Now we add $\chi(G)$ dimensions to make the representation touching again, and to ensure some space for the clique boxes $h^\varepsilon(C)$. Formally we define h_2 as

$$h_2(u)[i] = \begin{cases} h_1(u)[i] & \text{if } i \leq d \\ [1/5, 3/5] & \text{if } i > d \text{ and } c(u) < i - d \\ [0, 2/5] & \text{if } i > d \text{ and } c(u) = i - d \\ [2/5, 4/5] & \text{otherwise (if } c(u) > i - d > 0). \end{cases}$$

For any clique C of G , let $c(C)$ denote the color set $\{c(u) \mid u \in V(C)\}$. We now set

$$p_2(C)[i] = \begin{cases} p_1(C)[i] & \text{if } i \leq d \\ 2/5 & \text{if } i > d \text{ and } i - d \in c(C) \\ 1/2 & \text{otherwise.} \end{cases}$$

As h_2 is an extension of h_1 , and as in each dimension $j > d$, $h_2(v)[j]$ is an interval of length $2/5$ containing the point $2/5$ for every vertex v , we have that h_2 is an intersection representation of G by comparable boxes. To prove that it is touching consider two adjacent vertices u and v such that $c(u) < c(v)$, and let us note that $h_2(u)[d + c(u)] = [0, 2/5]$ and $h_2(v)[d + c(u)] = [2/5, 4/5]$.

For the \emptyset -clique-sum extendability, the **(vertices)** conditions are void. For the **(cliques)** conditions, since p_1 is chosen to be injective, the mapping p_2 is injective as well, implying that (c1) holds.

Consider now a clique C in G and a vertex $v \in V(G)$. If $c(v) \notin c(C)$, then $h_2(v)[c(v) + d] = [0, 2/5]$ and $p_2(C)[c(v) + d] = 1/2$, implying that $h_2^\varepsilon(C) \cap h_2(v) = \emptyset$. If $c(v) \in c(C)$ but $v \notin V(C)$, then letting $v' \in V(C)$ be the vertex of color $c(v)$, we have $vv' \notin E(G)$, and thus $h_1(v)$ is disjoint from $h_1(v')$. Since $p_1(C)$ is contained in the interior of $h_1(v')$, it follows that $h_2^\varepsilon(C) \cap h_2(v) = \emptyset$. Finally, suppose that $v \in C$. Since $p_1(C)$ is contained in the interior of $h_1(v)$, we have $h_2^\varepsilon(C)[i] \subset h_2(v)[i]$ for every $i \leq d$. For $i > d$ distinct from $d + c(v)$, we have $p_2^\varepsilon(C)[i] \in \{2/5, 1/2\}$ and $[2/5, 3/5] \subseteq h_2(v)[i]$, and thus $h_2^\varepsilon(C)[i] \subset h_2(v)[i]$. For $i = d + c(v)$, we have $p_2^\varepsilon(C)[i] = 2/5$ and $h_2(v)[i] = [0, 2/5]$, and thus $h_2^\varepsilon(C)[i] \cap h_2(v)[i] = \{p_2^\varepsilon(C)[i]\}$. Therefore, (c2) holds. \blacktriangleleft

Together, the preceding lemmas show that comparable box dimension is almost preserved by full clique-sums.

► **Corollary 13.** *Let \mathcal{G} be a class of graphs of chromatic number at most k . If \mathcal{G}' is the class of all graphs that can be obtained from \mathcal{G} by repeatedly performing full clique-sums, then $\dim_{cb}(\mathcal{G}') \leq \dim_{cb}(\mathcal{G}) + 2k$.*

Proof. Suppose a graph G is obtained from $G_1, \dots, G_m \in \mathcal{G}$ by a sequence of full clique-sums. Without loss of generality, the labelling of the graphs is chosen so that we first perform the full clique-sum on G_1 and G_2 , then on the resulting graph and G_3 , and so on. Let $C_1^* = \emptyset$ and for $i = 2, \dots, m$, let C_i^* be the root clique of G_i on which it is glued in the full clique-sum operation. By Lemmas 12 and 11, G_i has a C_i^* -clique-sum extendable touching representation by comparable boxes in \mathbb{R}^d , where $d = \dim_{cb}(\mathcal{G}) + 2k$. Repeatedly applying Lemma 10, we conclude that $\dim_{cb}(G) \leq d$. \blacktriangleleft

Putting this corollary together with Lemmas 3 and 6, we obtain the following bounds.

- **Corollary 14.** *Let \mathcal{G} be a class of graphs of comparable box dimension at most d .*
- *The class \mathcal{G}' of graphs obtained from \mathcal{G} by repeatedly performing full clique-sums has comparable box dimension at most $d + 2 \cdot 3^d$.*
 - *The closure of \mathcal{G}' by taking subgraphs has comparable box dimension at most 1250^d .*

Proof. The former bound directly follows from Corollary 13 and the bound on the chromatic number from Lemma 3. For the latter, we need to bound the star chromatic number of \mathcal{G}' . Suppose a graph G is obtained from $G_1, \dots, G_m \in \mathcal{G}$ by performing full clique-sums. For $i = 1, \dots, m$, suppose G_i has an acyclic coloring φ_i by at most k colors. Note that the vertices of any clique get pairwise different colors, and thus by permuting the colors, we can ensure that when we perform the full clique-sum, the vertices that are identified have the same color. Hence, we can define a coloring φ of G such that for each i , the restriction of φ to $V(G_i)$ is equal to φ_i . Let C be the union of any two color classes of φ . Then for each i , $G_i[C \cap V(G_i)]$ is a forest, and since $G[C]$ is obtained from these graphs by full clique-sums, $G[C]$ is also a forest. Hence, φ is an acyclic coloring of G by at most k colors. By [1], G has a star coloring by at most $2k^2 - k$ colors. Hence, Lemma 3 implies that \mathcal{G}' has star chromatic number at most $2 \cdot 25^d - 5^d$. The bound on the comparable box dimension of subgraphs of graphs from \mathcal{G}' then follows from Lemma 6. ◀

4 The strong product structure and minor-closed classes

A k -tree is any graph obtained by repeated full clique-sums on cliques of size k from cliques of size at most $k + 1$. A k -tree-grid is a strong product of a k -tree and a path. An *extended k -tree-grid* is a graph obtained from a k -tree-grid by adding at most k apex vertices. Dujmović et al. [4] proved the following result.

► **Theorem 15.** *Any graph G is a subgraph of the strong product of a k -tree-grid and K_m , where*

- *$k = 3$ and $m = 3$ if G is planar, and*
- *$k = 4$ and $m = \max(2g, 3)$ if G has Euler genus at most g .*

Moreover, for every t , there exists an integer k such that any K_t -minor-free graph G is a subgraph of a graph obtained by repeated clique-sums from extended k -tree-grids.

Let us first bound the comparable box dimension of a graph in terms of its Euler genus. As paths and m -cliques admit touching representations with hypercubes of unit size in \mathbb{R}^1 and in $\mathbb{R}^{\lceil \log_2 m \rceil}$ respectively, by Lemma 5 it suffices to bound the comparable box dimension of k -trees.

► **Theorem 16.** *For any k -tree G , $\dim_{cb}(G) \leq \dim_{cb}^{ext}(G) \leq k + 1$.*

Proof. Let H be a complete graph with $k + 1$ vertices and let C^* be a clique of size k in H . By Lemma 10, it suffices to show that H has a C^* -clique-sum extendable touching representation by hypercubes in \mathbb{R}^{k+1} . Let $V(C^*) = \{v_1, \dots, v_k\}$. We construct the representation h so that (v1) holds with $d_{v_i} = i$ for each i ; this uniquely determines the hypercubes $h(v_1), \dots, h(v_k)$. For the vertex $v_{k+1} \in V(H) \setminus V(C^*)$, we set $h(v_{k+1}) = [0, 1/2]^{k+1}$. This ensures that the (vertices) conditions holds.

38:10 On Comparable Box Dimension

For the **(cliques)** conditions, let us set the point $p(C)$ for every clique C as follows:

- $p(C)[i] = 0$ for every $i \leq k$ such that $v_i \in C$
- $p(C)[i] = \frac{1}{4}$ for every $i \leq k$ such that $v_i \notin C$
- $p(C)[k+1] = \frac{1}{2}$ if $v_{k+1} \in C$
- $p(C)[k+1] = \frac{3}{4}$ if $v_{k+1} \notin C$

By construction, it is clear that for each vertex $v \in V(H)$, $p(C) \in h(v)$ if and only if $v \in V(C)$.

For any two distinct cliques C_1 and C_2 , the points $p(C_1)$ and $p(C_2)$ are distinct. Indeed, by symmetry we can assume that for some i we have $v_i \in V(C_1) \setminus V(C_2)$, and this implies that $p(C_1)[i] < p(C_2)[i]$. Hence, the condition (c1) holds.

Consider now a vertex v_i and a clique C . As we observed before, if $v_i \notin V(C)$, then $p(C) \notin h(v_i)$, and thus $h^\varepsilon(C)$ and $h(v_i)$ are disjoint (for sufficiently small $\varepsilon > 0$). If $v_i \in C$, then the definitions ensure that $p(C)[i]$ is equal to the maximum of $h(v_i)[i]$, and that for $j \neq i$, $p(C)[j]$ is in $h(v_i)[j]$, implying that $h(v_i)[j] \cap h^\varepsilon(C)[j] = [p(C)[j], p(C)[j] + \varepsilon]$ for sufficiently small $\varepsilon > 0$. ◀

The *treewidth* $\text{tw}(G)$ of a graph G is the minimum k such that G is a subgraph of a k -tree. It is worth noting that the bound on the comparable box dimension of Theorem 16 actually extends to graphs of treewidth at most k (proof omitted).

► **Corollary 17.** *Every graph G satisfies $\text{dim}_{cb}(G) \leq \text{tw}(G) + 1$.*

As every planar graph G has a touching representation by cubes in \mathbb{R}^3 [11], we have that $\text{dim}_{cb}(G) \leq 3$. For graphs with higher Euler genus we can also derive upper bounds. Indeed, combining the previous observation on the representations of paths and K_m with Theorem 16, Lemma 5 and Corollary 7 we obtain:

► **Corollary 18.** *For every graph G of Euler genus g , there exists a supergraph G' of G such that $\text{dim}_{cb}(G') \leq 6 + \lceil \log_2 \max(2g, 3) \rceil$. Consequently,*

$$\text{dim}_{cb}(G) \leq 3 \cdot 81^7 \cdot \max(2g, 3)^{\log_2 81}.$$

Similarly, we can deal with proper minor-closed classes.

Proof of Theorem 1. Let \mathcal{G} be a proper minor-closed class. Since \mathcal{G} is proper, there exists t such that $K_t \notin \mathcal{G}$. By Theorem 15, there exists k such that every graph in \mathcal{G} is a subgraph of a graph obtained by repeated clique-sums from extended k -tree-grids. As we have seen, k -tree-grids have comparable box dimension at most $k + 2$, and by Lemma 4, extended k -tree-grids have comparable box dimension at most $2k + 2$. By Corollary 14, it follows that $\text{dim}_{cb}(\mathcal{G}) \leq 1250^{2k+2}$. ◀

Note that the graph obtained from K_{2n} by deleting a perfect matching has Euler genus $\Theta(n^2)$ and comparable box dimension n . It follows that the dependence of the comparable box dimension on the Euler genus cannot be subpolynomial (though the degree $\log_2 81$ of the polynomial established in Corollary 18 certainly can be improved). The dependence of the comparable box dimension on the size of the forbidden minor that we established is not explicit, as Theorem 15 is based on the structure theorem of Robertson and Seymour [17]. It would be interesting to prove Theorem 1 without using the structure theorem.

5 Fractional treewidth-fragility

Suppose G is a connected planar graph and v is a vertex of G . For an integer $k \geq 2$, give each vertex at distance d from v the color $d \bmod k$. Then deleting the vertices of any of the k colors results in a graph of treewidth at most $3k$. This fact (which follows from the result of Robertson and Seymour [18] on treewidth of planar graphs of bounded radius) is (in the modern terms) the basis of Baker's technique [2] for design of approximation algorithms. However, even quite simple graph classes, such as the strong products of three paths [3], do not admit such a coloring where the removal of any color class results in a graph of bounded treewidth. Nonetheless, a fractional version of this coloring concept is still very useful in the design of approximation algorithms [8] and applies to much more general graph classes, including all graph classes with strongly sublinear separators and bounded maximum degree [5].

A class of graphs \mathcal{G} is *fractionally treewidth-fragile* if there exists a function f such that for every graph $G \in \mathcal{G}$ and integer $k \geq 2$, there exist sets $X_1, \dots, X_m \subseteq V(G)$ such that each vertex belongs to at most m/k of them and $\text{tw}(G - X_i) \leq f(k)$ for every i (equivalently, there exists a probability distribution on the set $\{X \subseteq V(G) : \text{tw}(G - X) \leq f(k)\}$ such that $\Pr[v \in X] \leq 1/k$ for each $v \in V(G)$). For example, the class of planar graphs is (fractionally) treewidth-fragile, since we can let X_i consist of the vertices of color $i - 1$ in the coloring described at the beginning of the section.

It will be useful to have a different formulation of treewidth for the argument to follow. Recall that a *tree decomposition* of a graph G is a pair (T, β) , where T is a rooted tree and $\beta : V(T) \rightarrow 2^{V(G)}$ assigns a *bag* to each of its nodes, such that

- for each edge $uv \in E(G)$, there exists $x \in V(T)$ such that $u, v \in \beta(x)$, and
- for each vertex $v \in V(G)$, the set $\{x \in V(T) : v \in \beta(x)\}$ is non-empty and induces a connected subtree of T .

For nodes $x, y \in V(T)$, we write $x \preceq y$ if $x = y$ or x is a descendant of y in T . The *width* of the tree decomposition is the maximum of the sizes of the bags minus 1. The *treewidth* of a graph is the minimum of the widths of its tree decompositions. Let us remark that the treewidth obtained via this definition coincides with the one via k -trees of Section 4

The purpose of this section is to show that all graph classes of bounded comparable box dimension are fractionally treewidth-fragile. In fact, we prove this result in a more general setting, motivated by concepts from [6] and by applications to related representations. The argument is motivated by the idea used in the approximation algorithms for disk graphs by Erlebach et al. [10].

For a measurable set $A \subseteq \mathbb{R}^d$, let $\text{vol}(A)$ denote the Lebesgue measure of A . Given two measurable subsets A and B of \mathbb{R}^d and a positive integer s , we write $A \sqsubseteq_s B$ if for every $x \in B$, there exists a translation A' of A such that $x \in A'$ and $\text{vol}(A' \cap B) \geq \frac{1}{s} \text{vol}(A)$. Note that for two boxes A and B , we have $A \sqsubseteq_1 B$ if and only if $A \subseteq B$. An *s -comparable envelope representation* (ι, ω) of a graph G in \mathbb{R}^d consists of two functions $\iota, \omega : V(G) \rightarrow 2^{\mathbb{R}^d}$ such that for some ordering v_1, \dots, v_n of vertices of G ,

- for each i , $\omega(v_i)$ is a box, $\iota(v_i)$ is a measurable set, and $\iota(v_i) \subseteq \omega(v_i)$,
- if $i < j$, then $\omega(v_j) \sqsubseteq_s \iota(v_i)$, and
- if $i < j$ and $v_i v_j \in E(G)$, then $\omega(v_j) \cap \iota(v_i) \neq \emptyset$.

We say that the representation has *thickness at most t* if for every point $x \in \mathbb{R}^d$, there exist at most t vertices $v \in V(G)$ such that $x \in \iota(v)$. For example, if f is a touching representation of G by comparable boxes in \mathbb{R}^d , then (f, f) is a 1-comparable envelope representation of G in \mathbb{R}^d of thickness at most 2^d .

► **Theorem 19.** *For positive integers t , s , and d , the class of graphs with an s -comparable envelope representation in \mathbb{R}^d of thickness at most t is fractionally treewidth-fragile, with a function $f(k) = O_{t,s,d}(k^d)$.*

Proof. For a positive integer k , let $f(k) = (2ksd + 2)^d st$. Let (ι, ω) be an s -comparable envelope representation of a graph G in \mathbb{R}^d of thickness at most t , and let v_1, \dots, v_n be the corresponding ordering of the vertices of G . Let us define $\ell_{i,j} \in \mathbb{R}^+$ for $i = 1, \dots, n$ and $j \in \{1, \dots, d\}$ as an approximation of $ksd|\omega(v_i)[j]|$ such that $\ell_{i-1,j}/\ell_{i,j}$ is a positive integer. Formally, it is defined by the following process.

- Let $\ell_{1,j} = ksd|\omega(v_1)[j]|$.
- For $i = 2, \dots, n$, let $\ell_{i,j} = \ell_{i-1,j}$, if $\ell_{i-1,j} < ksd|\omega(v_i)[j]|$, and otherwise let $\ell_{i,j}$ be lowest fraction of $\ell_{i-1,j}$ that is greater than $ksd|\omega(v_i)[j]|$, formally $\ell_{i,j} = \min\{\ell_{i-1,j}/b \mid b \in \mathbb{N}^+ \text{ and } \ell_{i-1,j}/b \geq ksd|\omega(v_i)[j]|\}$.

Choose $x_j \in [0, \ell_{1,j}]$ uniformly at random, and let \mathcal{H}_j^i be the set of hyperplanes in \mathbb{R}^d consisting of the points whose j -th coordinate is equal to $x_j + m\ell_{i,j}$ for some $m \in \mathbb{Z}$. As $\ell_{i,j}$ is a multiple of $\ell_{i',j}$ whenever $i \leq i'$, we have that $\mathcal{H}_j^i \subseteq \mathcal{H}_j^{i'}$ whenever $i \leq i'$. For $i \in \{1, \dots, n\}$, the i -grid is $\mathcal{H}^i = \bigcup_{j=1}^d \mathcal{H}_j^i$, and we let the 0-grid $\mathcal{H}^0 = \emptyset$. Then, as above, we have that $\mathcal{H}^i \subseteq \mathcal{H}^{i'}$ whenever $i \leq i'$.

Let $X \subseteq V(G)$ consist of the vertices $v_a \in V(G)$ such that the box $\omega(v_a)$ intersects some hyperplane $H \in \mathcal{H}^a$, that is such that $x_j + m\ell_{a,j} \in \omega(v_a)[j]$, for some $j \in \{1, \dots, d\}$ and some $m \in \mathbb{Z}$. First, let us argue that $\Pr[v_a \in X] \leq 1/k$. Indeed, the set $[0, \ell_{1,j}] \cap \bigcup_{m \in \mathbb{Z}} (\omega(v_a)[j] - m\ell_{a,j})$ has measure $\frac{\ell_{1,j}}{\ell_{a,j}} \cdot |\omega(v_a)[j]|$, implying that for fixed j , this happens with probability $|\omega(v_a)[j]|/\ell_{a,j}$. Let a' be the largest integer such that $a' \leq a$ and $\ell_{a',j} < \ell_{a'-1,j}$ if such an index exists, and $a' = 1$ otherwise; note that $\ell_{a,j} = \ell_{a',j} \geq ksd|\omega(v_{a'})[j]|$. Moreover, since $\omega(v_a) \sqsubseteq_s \iota(v_{a'}) \subseteq \omega(v_{a'})$, we have $\omega(v_a)[j] \leq s\omega(v_{a'})[j]$. Combining these inequalities,

$$\frac{|\omega(v_a)[j]|}{\ell_{a,j}} \leq \frac{s\omega(v_{a'})[j]}{ksd|\omega(v_{a'})[j]|} = \frac{1}{kd}.$$

By the union bound, we conclude that $\Pr[v_a \in X] \leq 1/k$.

We now bound the treewidth of $G - X$. For $a \geq 0$, an a -cell is a maximal connected subset of $\mathbb{R}^d \setminus (\bigcup_{H \in \mathcal{H}^a} H)$. A set $C \subseteq \mathbb{R}^d$ is a cell if it is an a -cell for some $a \geq 0$. A cell C is *non-empty* if there exists $v \in V(G - X)$ such that $\iota(v) \subseteq C$. Note that there exists a rooted tree T whose vertices are the non-empty cells and such that for $x, y \in V(T)$, we have $x \preceq y$ if and only if $x \subseteq y$. For each non-empty cell C , define $\beta(C)$ to be the set of vertices $v_i \in V(G - X)$ such that $\iota(v) \cap C \neq \emptyset$ and C is an a -cell for some $a \geq i$.

Let us show that (T, β) is a tree decomposition of $G - X$. For each $v_j \in V(G - X)$, the j -grid is disjoint from $\omega(v_j)$, and thus $\iota(v_j) \subseteq \omega(v_j) \subset C$ for some j -cell $C \in V(T)$ and $v_j \in \beta(C)$. Consider now an edge $v_i v_j \in E(G - X)$, where $i < j$. We have $\omega(v_j) \cap \iota(v_i) \neq \emptyset$, and thus $\iota(v_i) \cap C \neq \emptyset$ and $v_i \in \beta(C)$. Finally, suppose that $v_j \in C'$ for some $C' \in V(T)$. Then C' is an a -cell for some $a \geq j$, and since $\iota(v_j) \cap C' \neq \emptyset$ and $\iota(v_j) \subset C$, we conclude that $C' \subseteq C$, and consequently $C' \preceq C$. Moreover, any cell C'' such that $C' \preceq C'' \preceq C$ (and thus $C' \subseteq C'' \subseteq C$) is an a' -cell for some $a' \geq j$ and $\iota(v_j) \cap C'' \supseteq \iota(v_j) \cap C' \neq \emptyset$, which implies that $v_j \in \beta(C'')$. It follows that $\{C' : v_j \in \beta(C')\}$ induces a connected subtree of T .

Finally, we bound the width of the decomposition (T, β) . Let C be a non-empty cell and let a be maximum number for which C is an a -cell. Then C is an open box with sides of lengths $\ell_{a,1}, \dots, \ell_{a,d}$. Consider $j \in \{1, \dots, d\}$:

- If $a = 1$, then $\ell_{a,j} = ksd|\omega(v_a)[j]|$.
- If $a > 1$ and $\ell_{a,j} = \ell_{a-1,j}$, then $\ell_{a,j} = \ell_{a-1,j} < 2ksd|\omega(v_a)[j]|$ (otherwise $\ell_{a,j} = \ell_{a-1,j}/b$ for some integer $b \geq 2$).

- If $a > 1$ and $\ell_{a,j} < \ell_{a-1,j}$, then $\ell_{a-1,j} \geq b \times ksd|\omega(v_a)[j]|$ for some integer $b \geq 2$. Now let b be the greatest such integer (that is such that $\ell_{a-1,j} < (b + 1) \times ksd|\omega(v_a)[j]|$) and note that

$$\ell_{a,j} = \frac{\ell_{a-1,j}}{b} < \frac{b+1}{b} ksd|\omega(v_a)[j]| < \frac{3}{2} ksd|\omega(v_a)[j]|.$$

Hence, in all cases we have $\ell_{a,j} < 2ksd|\omega(v_a)[j]|$. Let C' be the box with the same center as C and with $|C'[j]| = (2ksd + 2)|\omega(v_a)[j]|$. For any $v_i \in \beta(C) \setminus \{v_a\}$, we have $i \leq a$ and $\iota(v_i) \cap C \neq \emptyset$, and since $\omega(v_a) \sqsubseteq_s \iota(v_i)$, there exists a translation B_i of $\omega(v_a)$ that intersects $C \cap \iota(v_i)$ and such that $\text{vol}(B_i \cap \iota(v_i)) \geq \frac{1}{s} \text{vol}(\omega(v_a))$. Note that as B_i intersects C , we have that $B_i \subseteq C'$. Using the initial assumption that the representation has thickness at most t , we now have

$$\begin{aligned} \text{vol}(C') &\geq \text{vol}\left(C' \cap \bigcup_{v_i \in \beta(C) \setminus \{v_a\}} \iota(v_i)\right) \\ &\geq \text{vol}\left(\bigcup_{v_i \in \beta(C) \setminus \{v_a\}} B_i \cap \iota(v_i)\right) \\ &\geq \frac{1}{t} \sum_{v_i \in \beta(C) \setminus \{v_a\}} \text{vol}(B_i \cap \iota(v_i)) \\ &\geq \frac{\text{vol}(\omega(v_a))(|\beta(C)| - 1)}{st}. \end{aligned}$$

Since $\text{vol}(C') = (2ksd + 2)^d \text{vol}(\omega(v_a))$, it follows that

$$|\beta(C)| - 1 \leq (2ksd + 2)^d st = f(k),$$

as required. ◀

The proof that (generalizations of) graphs with bounded comparable box dimensions have sublinear separators in [6] is indirect; it is established that these graphs have polynomial coloring numbers, which in turn implies they have polynomial expansion, which then gives sublinear separators using the algorithm of Plotkin, Rao, and Smith [16]. The existence of sublinear separators is known to follow more directly from fractional treewidth-fragility. Indeed, since $\Pr[v \in X] \leq 1/k$, there exists $X \subseteq V(G)$ such that $\text{tw}(G - X) \leq f(k)$ and $|X| \leq |V(G)|/k$. The graph $G - X$ has a balanced separator of size at most $\text{tw}(G - X) + 1$, which combines with X to a balanced separator of size at most $|V(G)|/k + f(k) + 1$ in G . Optimizing the value of k (choosing it so that $|V(G)|/k = f(k)$), we obtain the following corollary of Theorem 19.

► **Corollary 20.** *For positive integers t, s , and d , every graph G with an s -comparable envelope representation in \mathbb{R}^d of thickness at most t has a sublinear separator of size $O_{t,s,d}(|V(G)|^{\frac{d}{d+1}})$.*

References

- 1 Michael O Albertson, Glenn G Chappell, Hal A Kierstead, André Kündgen, and Radhika Ramamurthi. Coloring with no 2-colored p_4 's. *the electronic journal of combinatorics*, pages R26–R26, 2004.
- 2 B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM (JACM)*, 41(1):153–180, 1994.

- 3 E. Berger, Z. Dvořák, and S. Norin. Treewidth of grid subsets. *Combinatorica*, 2017. Accepted, doi.org/10.1007/s00493-017-3548-5.
- 4 V. Dujmović, G. Joret, P. Micek, P. Morin, T. Ueckerdt, and D. R. Wood. Planar graphs have bounded queue-number. *Journal of the ACM*, 67:22, 2020.
- 5 Z. Dvořák. Sublinear separators, fragility and subexponential expansion. *European Journal of Combinatorics*, 52:103–119, 2016.
- 6 Z. Dvořák, R. McCarty, and S. Norin. Sublinear separators in intersection graphs of convex shapes. *arXiv*, 2001.01552, 2020. arXiv:2001.01552.
- 7 Zdeněk Dvořák. Approximation metatheorem for fractionally treewidth-fragile graphs. *arXiv*, 2103.08698, 2021. arXiv:2103.08698.
- 8 Zdeněk Dvořák and Abhiruk Lahiri. Approximation schemes for bounded distance problems on fractionally treewidth-fragile graphs. In *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference)*, volume 204 of *LIPICs*, pages 40:1–40:10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- 9 Zdeněk Dvořák, Jakub Pekárek, Torsten Ueckerdt, and Yelena Yuditsky. Weak coloring numbers of intersection graphs. *arXiv*, 2103.17094, 2021. arXiv:2103.17094.
- 10 Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34:1302–1323, 2005.
- 11 Stefan Felsner and Mathew C Francis. Contact representations of planar graphs with cubes. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 315–320, 2011.
- 12 Martin Grohe, Stephan Kreutzer, Roman Rabinovich, Sebastian Siebertz, and Konstantinos Stavropoulos. Coloring and covering nowhere dense graphs. *SIAM Journal on Discrete Mathematics*, 32:2467–2481, 2018.
- 13 P. Koebe. Kontaktprobleme der Konformen Abbildung. *Math.-Phys. Kl.*, 88:141–164, 1936.
- 14 L. Lovász. *Graphs and Geometry*. American Mathematical Society, Providence, 2019.
- 15 J. Nešetřil and P. Ossona de Mendez. *Sparsity (Graphs, Structures, and Algorithms)*, volume 28 of *Algorithms and Combinatorics*. Springer, 2012.
- 16 Serge Plotkin, Satish Rao, and Warren D Smith. Shallow excluded minors and improved graph decompositions. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 462–470. Society for Industrial and Applied Mathematics, 1994.
- 17 N. Robertson and P. D. Seymour. Graph Minors. XVI. Excluding a non-planar graph. *J. Combin. Theory, Ser. B*, 89(1):43–76, 2003.
- 18 Neil Robertson and Paul D. Seymour. Graph Minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B*, 36:49–64, 1984.
- 19 Horst Sachs. Coin graphs, polyhedra, and conformal mapping. *Discrete Mathematics*, 134:133–138, 1994.

Weak Coloring Numbers of Intersection Graphs

Zdeněk Dvořák  

Charles University, Prague, Czech Republic

Jakub Pekárek 

Charles University, Prague, Czech Republic

Torsten Ueckerdt 

Karlsruhe Institute of Technology, Germany

Yelena Yuditsky  

Université Libre de Bruxelles, Brussels, Belgium

Abstract

Weak and strong coloring numbers are generalizations of the degeneracy of a graph, where for a positive integer k , we seek a vertex ordering such that every vertex can (weakly respectively strongly) reach in k steps only few vertices that precede it in the ordering. Both notions capture the sparsity of a graph or a graph class, and have interesting applications in structural and algorithmic graph theory. Recently, Dvořák, McCarty, and Norin observed a natural volume-based upper bound for the strong coloring numbers of intersection graphs of well-behaved objects in \mathbb{R}^d , such as homothets of a compact convex object, or comparable axis-aligned boxes.

In this paper, we prove upper and lower bounds for the k -th weak coloring numbers of these classes of intersection graphs. As a consequence, we describe a natural graph class whose strong coloring numbers are polynomial in k , but the weak coloring numbers are exponential. We also observe a surprising difference in terms of the dependence of the weak coloring numbers on the dimension between touching graphs of balls (single-exponential) and hypercubes (double-exponential).

2012 ACM Subject Classification Mathematics of computing → Combinatoric problems; Mathematics of computing → Graph coloring

Keywords and phrases geometric intersection graphs, weak and strong coloring numbers

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.39

Funding *Zdeněk Dvořák*: Supported by the ERC-CZ project LL2005 (Algorithms and complexity within and beyond bounded expansion) of the Ministry of Education of Czech Republic.

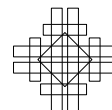
Jakub Pekárek: Supported by the ERC-CZ project LL2005 (Algorithms and complexity within and beyond bounded expansion) of the Ministry of Education of Czech Republic.

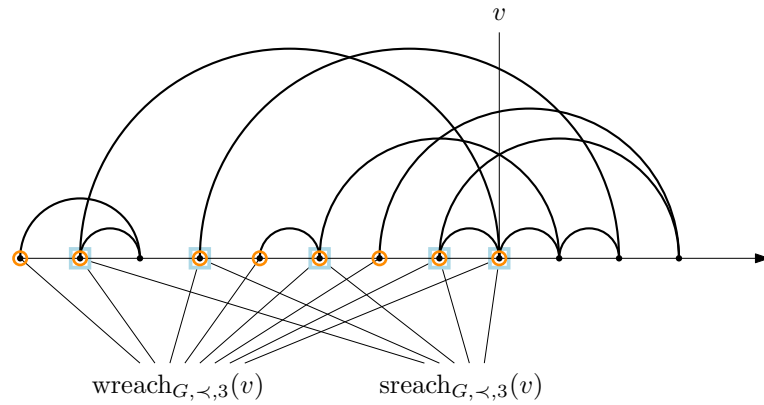
Acknowledgements This research was carried out at the workshop on Generalized Coloring Numbers organized by Michał Pilipczuk and Piotr Micek in February 2021. We would like to thank the organizers and all participants for creating a friendly and productive environment. Special thanks go to Stefan Felsner for fruitful discussions.

1 Introduction

It is well known that if every subgraph of a graph G has average degree at most d , then G is d -degenerate, that is, there exists a linear ordering of the vertices of G such that each vertex has at most d neighbors that precede it in the ordering. Conversely, every subgraph of a d -degenerate graph has average degree at most $2d$. This fact is often used in design of algorithms for sparse graphs, where a result is obtained by processing the vertices one by one in the degeneracy ordering.

For algorithmic problems that involve interactions over larger distances, a stronger notion of sparsity is needed. Such a notion of *bounded expansion* was developed by Nešetřil and Ossona de Mendez [12] and can be formulated in terms of the dependence of the density of





■ **Figure 1** A vertex ordering \prec of a graph G , and the sets $\text{wreach}_{G, \prec, k}(v)$ and $\text{sreach}_{G, \prec, k}(v)$ for a vertex v and $k = 3$.

minors or topological minors that appear in the considered graphs on the depths of these minors (we do not give a precise definition since it is somewhat technical and we do not need it in this paper). As was shown by Zhu [15], there is also an equivalent degeneracy-like characterization of bounded expansion, in terms of generalized coloring numbers, that is weak and strong coloring numbers defined below. The generalized coloring numbers were previously introduced by Kierstead and Yang [10] in the context of marking and coloring games on graphs.

Given a linear ordering \prec of the vertices of a graph G and an integer $k \geq 0$, a vertex u is *weakly k -reachable* from a vertex v if $u \preceq v$ and there exists a path in G from v to u of length at most k with all internal vertices greater than u in \prec , and *strongly k -reachable* if there exists such a path with all internal vertices greater than v in \prec ; see Figure 1 for an illustration. Let $\text{wreach}_{G, \prec, k}(v)$ and $\text{sreach}_{G, \prec, k}(v)$ denote the sets of vertices that are weakly and strongly k -reachable from v , respectively. We define *weak and strong coloring numbers* for a given ordering \prec as

$$\begin{aligned} \text{wcol}_{\prec, k}(G) &= \max_{v \in V(G)} |\text{wreach}_{G, \prec, k}(v)| \\ \text{scol}_{\prec, k}(G) &= \max_{v \in V(G)} |\text{sreach}_{G, \prec, k}(v)| \end{aligned}$$

The weak and strong coloring numbers of a graph are then obtained by minimizing over all linear orderings of $V(G)$.

$$\begin{aligned} \text{wcol}_k(G) &= \min_{\prec} \text{wcol}_{\prec, k}(G) \\ \text{scol}_k(G) &= \min_{\prec} \text{scol}_{\prec, k}(G) \end{aligned}$$

Note that for $k = 1$, both $\text{wreach}_{G, \prec, 1}(v) \setminus \{v\}$ and $\text{sreach}_{G, \prec, 1}(v) \setminus \{v\}$ consist of the neighbors of v that precede it in the ordering \prec , and thus $\text{scol}_1(G) = \text{wcol}_1(G)$ coincide with the *coloring number* of the graph G , equal to the degeneracy of G plus one.

1.1 Properties and applications of generalized coloring numbers

The following basic claims can be found for example in [12]. One can easily check that both $\text{wcol}_k(G)$ and $\text{scol}_k(G)$ are non-decreasing in k and that $\text{scol}_k(G) \leq \text{wcol}_k(G) \leq (\text{scol}_k(G))^k$ for any positive integer k . Moreover, for every $k \geq |V(G)|$, $\text{scol}_k(G)$ is equal to the treewidth

of G and $wcol_k(G)$ is equal to the treedepth of G . A greedy coloring algorithm applied along the corresponding vertex ordering shows that the chromatic number of G is at most $scol_1(G) = wcol_1(G)$, the acyclic chromatic number of G is at most $scol_2(G)$, and the star chromatic number of G is at most $wcol_2(G)$.

Algorithmic applications of the generalized coloring numbers include for example:

- Generating sparse neighborhood covers used in decision algorithms for problems expressible in the first-order logic [8].
- Constant-factor approximation for distance versions of domination number and independence number [2], with further applications in fixed-parameter algorithms and kernelization [5].
- Practical algorithm for counting the number of appearances of fixed subgraphs [13].

As we mentioned before, Zhu [15] proved that generalized coloring numbers are bounded exactly for graph classes with bounded expansion (which include planar graphs and more generally all proper classes closed under taking minors or topological minors, graphs with bounded maximum degree, graphs that can be drawn in the plane with a bounded number of crossings per edge, intersection graphs of balls with bounded clique number, and many others). More precisely, for any class \mathcal{G} with bounded expansion, there exist functions $f_{\mathcal{G}}^s$ and $f_{\mathcal{G}}^w$ such that for every graph $G \in \mathcal{G}$ and every positive integer k , we have $scol_k(G) \leq f_{\mathcal{G}}^s(k)$ and $wcol_k(G) \leq f_{\mathcal{G}}^w(k)$. However, the general bounds arising from Zhu’s result are rather weak, and since the time complexity of the aforementioned algorithms depends on the generalized coloring numbers, we are interested in more precise bounds for specific graph classes.

1.2 Bounds on generalized coloring numbers

Quite a bit is known about the maximum possible values of generalized coloring numbers of many natural graph classes, as summarized in the following table:

Class	$scol_k$	$wcol_k$
treewidth $\leq t$	$t + 1$ [7]	$\binom{k+t}{t}$ [7]
outerplanar	3	$\Theta(k \log k)$ [9]
planar	$\Theta(k)$ [14]	$\Omega(k^2 \log k)$ [9] $O(k^3)$ [14]
genus g	$O(gk)$ [14]	$O(gk + k^3)$ [14]
no K_t minor	$O(t^2 k)$ [14]	$\Omega(k^{t-2})$ [7] $O(k^{t-1})$ [14]
no K_t topological minor	$\Omega((t - 3)^{k/4})$ [6, attributed to Norin]	$t^{O(k)}$ [7]

Moreover, Dvořák et al. [3] observed that in many classes of intersection graphs of geometric objects in \mathbb{R}^d , a non-increasing ordering of the objects according to their volume easily implies that their strong coloring number is at most $O(k^d)$. The starting point of this paper is the investigation of the same ordering from the perspective of the weak coloring numbers.

1.3 Strong coloring numbers of intersection graphs

Let S be a finite set of subsets of \mathbb{R}^d , which we call *objects*. The *intersection graph* of S is the graph G with $V(G) = S$ and with $uv \in E(G)$ if and only if $u \cap v \neq \emptyset$. For an integer $t \geq 1$, we say that the set S is *t-thin* if every point of \mathbb{R}^d is contained in the interior of at most t objects from S ; in the case $t = 1$, we say S is a *touching representation* of G . For example, a famous result of Koebe [11] states that a graph is planar if and only if it has

39:4 Weak Coloring Numbers of Intersection Graphs

a touching representation by balls in \mathbb{R}^2 . Another example can be found in [4], where it is shown that the graphs in any proper minor-closed class have touching representation by *comparable* axis-aligned boxes in bounded dimension. That is, by a set S of axis-aligned boxes which has the additional property that for every $u, v \in S$, a translation of u is a subset of v or vice versa. As observed in [3], there is a very natural way of bounding the strong coloring numbers for thin intersection graphs of certain classes of objects by ordering the vertices in a non-increasing order according to the size of the objects that represent the vertices. Note that, by the definitions of the coloring numbers, if it is possible to show an upper bound on the strong coloring number in this ordering (or any ordering) then it implies an upper bound on the strong coloring number of the intersection graph. In particular, this approach works in the case the objects in S are

- scaled and translated copies of the same centrally symmetric compact convex object (this includes intersection graphs of balls and of axis-aligned hypercubes); or
- *b-ball-like* for some real number $b \geq 1$, i.e., every $v \in S$ is a compact convex set satisfying $\text{vol}(v) \geq \text{vol}(B(\text{diam}(v)/2))/b$, where $B(a)$ is the ball in \mathbb{R}^d of radius a , $\text{diam}(v)$ is the maximum distance between any two points of v , and $\text{vol}(v)$ is the volume of v ; or
- comparable axis-aligned boxes.

As we are going to build on this argument, let us give a sketch of it. A linear ordering \prec of a finite set of compact objects S is *size-wise* if for all $u, v \in S$ such that $u \prec v$, we have $\text{diam}(u) \geq \text{diam}(v)$. Roughly, the idea behind the proof of the next lemma is that in a size-wise ordering, the number of objects it is possible to strongly k -reach from a given object v , is bounded by the maximum order of a t -thin system of objects of larger size which can be placed in a scaled instance of v .

► **Lemma 1.** *Let d and t be positive integers. Let S be a t -thin finite set of compact convex objects in \mathbb{R}^d and let G be the intersection graph of S . Let \prec be a size-wise linear ordering of S . For each integer $k \geq 1$,*

- (a) *if S consists of scaled and translated copies of the same centrally symmetric object, or if S is a set of comparable axis-aligned boxes, then $\text{scol}_{\prec, k}(G) \leq t(2k + 1)^d$, and*
- (b) *if S consists of b -ball-like objects for a real number $b \geq 1$, then $\text{scol}_{\prec, k}(G) \leq bt(2k + 2)^d$.*

Proof. Consider a vertex $v \in V(G)$; we need to provide an upper bound on $|\text{sreach}_{G, \prec, k}(v)|$. For any $m \geq 0$, in case (a) let $B_m(v)$ be the object obtained by scaling v by the factor of $2m + 1$, with the center p of v being the fixed point; i.e., $B_m(v) = \{p + (2m + 1)(q - p) : q \in v\}$. In case (b), let $B_m(v)$ be a ball of radius $(m + 1) \text{diam}(v)$ centered at an arbitrarily chosen point of v .

For each $u \in \text{sreach}_{G, \prec, k}(v)$, observe that $u \cap B_{k-1}(v) \neq \emptyset$, as u is joined to v through a path with at most $k - 1$ internal vertices, each represented by an object smaller or equal to v in size. In case (a), observe that there exists a translation u' of v such that $u' \subseteq u$ and $u' \cap B_{k-1}(v) \neq \emptyset$. In case (b), let u' be a scaled translation of u such that $u' \subseteq u$, $u' \cap B_{k-1}(v) \neq \emptyset$, and $\text{diam}(u') = \text{diam}(v)$. Note that in the former case we have $\text{vol}(u') = \text{vol}(v) = (2k + 1)^{-d} \text{vol}(B_k(v))$, and in the latter case we have

$$\begin{aligned} \text{vol}(u') &= \frac{\text{diam}^d(v)}{\text{diam}^d(u)} \text{vol}(u) \geq \frac{\text{diam}^d(v)}{b \text{diam}^d(u)} \text{vol}(B(\text{diam}(u)/2)) \\ &= b^{-1} \text{vol}(B(\text{diam}(v)/2)) = b^{-1}(2k + 2)^{-d} \text{vol}(B_k(v)). \end{aligned}$$

In either case, observe that $u' \subseteq B_k(v)$, and since S is t -thin, we have

$$\sum_{u \in \text{sreach}_{G, \prec, k}(v)} \text{vol}(u') \leq t \text{vol}(B_k(v)).$$

Therefore, $|\text{sreach}_{G, \prec, k}(v)| \leq t(2k + 1)^d$ in case (a) and $|\text{sreach}_{G, \prec, k}(v)| \leq bt(2k + 2)^d$ in case (b). ◀

That is, the strong coloring numbers of these graph classes are polynomial in k , with a uniform ordering of vertices that works for all values of k . For weak coloring numbers, a general upper bound is as follows.

► **Observation 2.** For any graph G , a linear ordering \prec of its vertices, and an integer $k \geq 1$,

$$\text{wcol}_{\prec, k}(G) \leq \sum_{i=1}^k \text{scol}_{\prec, i}(G) \text{wcol}_{\prec, k-i}(G).$$

In particular, if there exists $c > 1$ such that $\text{scol}_{\prec, k}(G) \leq c^k$ for every $k \geq 1$, then $\text{wcol}_{\prec, k}(G) \leq (2c)^k$ for every $k \geq 1$.

For graphs from the classes described in Lemma 1, we obtain an exponential bound on the weak coloring numbers, more precisely $\text{wcol}_k(G) \leq (2t3^d)^k$ in case (a) and $\text{wcol}_k(G) \leq (2bt4^d)^k$ in case (b).

2 Our results

Joret and Wood (see [6]) conjectured that every class of graphs with polynomial strong coloring numbers also has polynomial weak coloring numbers (more precisely, this claim is implied by their conjecture regarding weak coloring numbers of graphs of polynomial expansion). This turns out not to be the case; Grohe et al. [7] showed that the class of graphs obtained by subdividing all edges of each graph the number of times equal to its treewidth has superpolynomial weak coloring numbers, while their strong coloring numbers are linear. However, one could still expect this conjecture to hold for “natural” graph classes, and thus we ask whether the weak coloring numbers are polynomial for the graph classes described in Lemma 1. On the positive side, we obtain the following result.

► **Theorem 3.** Let d and t be positive integers. Let S be a t -thin finite set of compact convex objects in \mathbb{R}^d and let G be the intersection graph of S . Let \prec be a sizewise linear ordering of S . For each integer $k \geq 1$:

(a) If S consists of scaled and translated copies of the same centrally symmetric object, then

$$\text{wcol}_{\prec, k}(G) \leq t \max(1, \lceil \log_2 k \rceil) (4k - 1)^d \binom{k + t5^d + 2}{t5^d + 2}.$$

(b) If S consists of b -ball-like objects for a real number $b \geq 1$, then

$$\text{wcol}_{\prec, k}(G) \leq tb \max(1, \lceil \log_2 k \rceil) (4k)^d \binom{k + tb6^d + 2}{tb6^d + 2}.$$

Moreover, there exists k_0 (depending only on d) such that if S consists of balls, then for every $k \geq k_0$,

$$\text{wcol}_{\prec, k}(G) \leq t \max(1, \lceil \log_2 k \rceil) (4k - 1)^d \binom{k + 2t + 2}{2t + 2}.$$

Asymptotically, the bounds in (a) and (b) in the above theorem are doubly exponential in the dimension d and singly exponential in t (and b), and for fixed d and t , they depend on k polynomially. Note that the bounds are for the full weak coloring numbers (minimized over all orderings), not just with respect to the sizewise ordering. Theorem 3 is qualitatively tight in several surprising aspects, summarized in the following result.

► **Theorem 4.** *For every positive integer k :*

- (i) *There exists a touching graph F_k of comparable axis-aligned boxes in \mathbb{R}^3 such that $\text{wcol}_{2k}(F_k) \geq 2^{k+1} - 1$.*
- (ii) *For every t , there exists a t -thin set of axis-aligned squares in \mathbb{R}^2 whose intersection graph $H_{k,t}$ satisfies $\text{wcol}_{2k}(H_{k,t}) \geq \binom{k+t}{t}$.*
- (iii) *For every $d \geq 1$, the graph $H_{k,2^d-1}$ can also be represented as a touching graph of axis-aligned hypercubes in \mathbb{R}^{d+2} .*

That is:

- (i) The class of touching graphs of comparable axis-aligned boxes in \mathbb{R}^3 has polynomial strong coloring numbers by Lemma 1, but exponential weak coloring numbers by Theorem 4(i). This provides a rather natural counterexample to the conjecture of Joret and Wood.

Let us remark that touching graphs of rectangles in \mathbb{R}^2 are obtained from planar graphs by adding crossing edges into faces of size four (when four of the boxes share corners), and such graphs have polynomial weak coloring numbers (this follows e.g. from their product structure [1]). Hence, the dimension three in the previous claim cannot be decreased.

- (ii) Lemma 1 shows that the strong coloring numbers depend linearly on the thinness t of the representation, while the bounds on the weak coloring numbers in Theorem 3 contain t in the exponent. As shown in Theorem 4(ii), in dimension at least two this cannot be avoided (if we want a bound which is not exponential in k) and Theorem 3 cannot be strengthened so that only the multiplicative constant would depend on t .

Let us also remark that t -thin intersection graphs of intervals in \mathbb{R} are interval graphs of clique number at most $2t$. As was pointed to us by Gwenaël Joret, any interval graph of clique number ω satisfies $\text{wcol}_k(G) \leq \binom{\omega+1}{2}(k+1)$, as shown by an ordering obtained by placing first the vertices of a maximal system of pairwise disjoint cliques of size ω and then recursively processing the remainder of the graph which has clique number smaller than ω . Hence, the dimension two in the previous claim cannot be decreased.

- (iii) In the case (a) of Theorem 3, and in particular for the touching graphs of axis-aligned hypercubes, the exponent must be exponential in the dimension, in a contrast to the case of touching graphs of balls.

3 Upper bounds

In order to prove Theorem 3 for all the classes at once, let us formulate an abstract graph property $P(f, a, e)$ on which the proof is based. For a graph G , a function $r: V(G) \rightarrow \mathbb{R}^+$ and $u, v \in V(G)$, let us define $\lambda_r(u, v)$ as the minimum of $\sum_{x \in V(Q) \setminus \{u, v\}} r(x)$ over all paths Q from u to v in G . For a function $f: \mathbb{Z}_0^+ \rightarrow \mathbb{Z}^+$ and positive integers a and e , we say that (G, r) has the property $P(f, a, e)$ if

- (i) for each $v \in V(G)$ and integers $s \geq 1$ and $p \geq 0$, there are at most $f(p)$ vertices $u \in V(G)$ such that $r(u) \geq sr(v)$ and $\lambda_r(u, v) \leq psr(v)$, and
- (ii) for each $v \in V(G)$ and each positive integer s , every sequence u_1, u_2, \dots of distinct vertices of G such that $\lambda_r(u_i, v) \leq sr(v)$ and $r(u_i) \geq a^i sr(v)$ for each i has length at most e .

Let us remark that $P(f, a, e)$ implies $P(f, a', e)$ for every $a' \geq a$, and (i) implies (ii) with $a = 1$ and $e = f(1)$. The following lemma is proved similarly to Lemma 1. In the lemma, for the role of the function r , we use diam . Intuitively, part (i) says that the number of objects with large diam that can be reached with a path with bounded diam from some object v is bounded. Part (ii) says that the number of objects with an increasing diam that can reach an object v with a path of bounded diam is also bounded.

► **Lemma 5.** *Let d and t be positive integers. Let S be a t -thin finite set of compact convex objects in \mathbb{R}^d and let G be the intersection graph of S . For $v \in V(G)$, let $r(v) = \text{diam}(v)$.*

- (a) *If S consists of scaled and translated copies of the same centrally symmetric object, then (G, r) has the property $P(p \mapsto t(2p + 3)^d, 1, t5^d)$.*
- (b) *If S consists of b -ball-like objects for $b \geq 1$, then (G, r) has the property $P(p \mapsto tb(2p + 4)^d, 1, tb6^d)$.*
- (c) *If S consists of balls, then there exists a such that (G, r) has the property $P(p \mapsto t(2p + 3)^d, a, 2t)$.*

Proof. Consider a vertex $v \in V(G)$ and integers $s \geq 1$ and $p \geq 0$. For any $m \geq 0$, in cases (a) and (c) let $B_m(v)$ be the object obtained by scaling v by the factor of $2m + 1$, with the center of v being the fixed point. In case (b), let $B_m(v)$ be a ball of radius $(m + 1) \text{diam}(v)$ centered at an arbitrarily chosen point of v . Let U be the set of vertices $u \in V(G)$ such that $r(u) \geq sr(v)$ and $\lambda_r(u, v) \leq psr(v)$. Observe that for any $u \in U$, we have $u \cap B_{ps}(v) \neq \emptyset$. Let u' be a scaled translation of u such that $u' \subseteq u$, $u' \cap B_{ps}(v) \neq \emptyset$, and $\text{diam}(u') = s \text{diam}(v)$. For each $m \geq 0$, in cases (a) and (c), we have

$$\text{vol}(u') = s^d \text{vol}(v) = \left(\frac{s}{2m+1}\right)^d \text{vol}(B_m(v)),$$

and in case (b) we have

$$\text{vol}(u') \geq b^{-1} s^d \text{vol}(B(\text{diam}(v)/2)) = b^{-1} \left(\frac{s}{2m+2}\right)^d \text{vol}(B_m(v)).$$

In either case, we have $u' \subseteq B_{(p+1)s}(v)$, and since S is t -thin, it follows that

$$|U| \leq t \left(\frac{2(p+1)s+1}{s}\right)^d \leq t(2p + 3)^d$$

in cases (a) and (c), and

$$|U| \leq tb \left(\frac{2(p+1)s+2}{s}\right)^d \leq tb(2p + 4)^d$$

in case (b). Hence, the part (i) of the property $P(f, a, e)$ is verified, and by the observations made before the lemma, this finishes the proof for the cases (a) and (b).

Let us now consider the part (ii) in case (c). Let Q be a half-space whose boundary hyperplane touches $B_s(v)$ and is otherwise disjoint from $B_s(v)$. There exists l such that $\text{vol}(Q \cap B_{ls}(v)) \geq \left(\frac{1}{2} - \frac{1}{6t}\right) \text{vol}(B_{ls}(v))$; let us fix smallest such l . For $a \geq 1$, let C_a be a ball touching $B_s(v)$ of radius $as \text{rad}(v)$. I.e. $C_a \subseteq Q$. Note that

$$\lim_{a \rightarrow \infty} \frac{\text{vol}(C_a \cap B_{ls}(v))}{\text{vol}(B_{ls}(v))} = \frac{\text{vol}(Q \cap B_{ls}(v))}{\text{vol}(B_{ls}(v))},$$

and thus there exists a such that $\text{vol}(C_a \cap B_{ls}(v)) \geq \left(\frac{1}{2} - \frac{1}{5t}\right) \text{vol}(B_{ls}(v))$; let us fix smallest such a .

39:8 Weak Coloring Numbers of Intersection Graphs

Consider a sequence u_1, u_2, \dots, u_n of distinct vertices of G such that $\lambda_r(u_i, v) \leq sr(v)$ and $r(u_i) \geq a^i sr(v)$ for each i . In particular, note that $\text{rad}(u_i) \geq \text{rad}(C_a)$ for each i . From the observation made in the first paragraph of the proof, we have $u_i \cap B_s(v) \neq \emptyset$, and it follows that

$$\frac{\text{vol}(u_i \cap B_{1s}(v))}{\text{vol}(B_{1s}(v))} \geq \frac{\text{vol}(C_a \cap B_{1s}(v))}{\text{vol}(B_{1s}(v))} \geq \frac{1}{2} - \frac{1}{5t}.$$

Since S is t -thin and n is an integer, this implies $n \leq 2t$, verifying the part (ii) of the property $P(p \mapsto t(2p+3)^d, a, 2t)$. \blacktriangleleft

To bound the weak coloring numbers, we need the following result about graphs of bounded pathwidth which appears in a stronger form (for treewidth) in van den Heuvel et al. [14]. For us, it is convenient to state the result as follows (without explicitly defining pathwidth), and thus we include the proof for completeness. A path $P = v_1 v_2 \dots v_m$ in a graph G with a linear ordering \prec of vertices is *decreasing* if $v_1 \succ v_2 \succ \dots \succ v_m$. For each $v \in V(G)$, we define $\text{decr}_{G, \prec, k}(v)$ as the set of vertices reachable from v by decreasing paths of length at most k .

► Lemma 6. *Let k and w be non-negative integers. Let \prec be a linear ordering of the vertices of a graph G . If for every $x \in V(G)$, at most w vertices $y \prec x$ have a neighbor $y' \succeq x$, then $|\text{decr}_{G, \prec, k}(v)| \leq \binom{k+w}{w}$ for every $v \in V(G)$.*

Proof. Without loss of generality, we assume that if $yy' \in E(G)$ and $y \prec y'$, then y is also adjacent to all vertices x such that $y \prec x \prec y'$. Indeed, adding such an edge yx does not violate the assumptions and can only increase $|\text{decr}_{G, \prec, k}(v)|$.

The proof is by induction on $k+w$. Note that $|\text{decr}_{G, \prec, 0}(v)| = 1$, and thus we can assume $k \geq 1$. If no neighbor of v is smaller than v , then $|\text{decr}_{G, \prec, k}(v)| = 1$, and thus the claim of the lemma holds. Hence, we can assume v has such a neighbor, and in particular $w \geq 1$. Let z be the smallest neighbor of v . Let G' be the subgraph of G induced by the vertices greater than z and smaller or equal to v . Since z is adjacent to all the vertices of G' , then for each $x \in V(G')$, at most $w-1$ vertices $y \prec x$ of G' have a neighbor $y' \succeq x$ in G' .

Consider now a vertex $u \in \text{decr}_{G, \prec, k}(v)$, and let Q be a decreasing path of length at most k from v to u . If $z \prec u$, then Q is also a decreasing path in G' , and thus $u \in \text{decr}_{G', \prec, k}(v)$. Note that $|\text{decr}_{G', \prec, k}(v)| \leq \binom{k+w-1}{w-1}$ by the induction hypothesis. If $u \prec z$, consider the edge $u'z'$ of Q such that $u' \prec z$ and $z \preceq z'$. Note that u' is not adjacent to v by the minimality of z , and thus $z' \neq v$. Moreover, by the assumption made in the first paragraph, $u'z \in E(G)$. Hence, u is reachable from v by the decreasing path of length at most k starting with vzu' and continuing along Q , and thus $u \in \text{decr}_{G, \prec, k-1}(z)$. If $u = z$, then we also have $u \in \text{decr}_{G, \prec, k-1}(z)$. By the induction hypothesis, we have $|\text{decr}_{G, \prec, k-1}(z)| \leq \binom{k+w-1}{w}$.

Therefore,

$$\begin{aligned} |\text{decr}_{G, \prec, k}(v)| &= |\text{decr}_{G', \prec, k}(v)| + |\text{decr}_{G, \prec, k-1}(z)| \\ &\leq \binom{k+w-1}{w-1} + \binom{k+w-1}{w} = \binom{k+w}{w}. \end{aligned} \quad \blacktriangleleft$$

We use the following corollary, obtained by applying Lemma 6 to the graph obtained by contracting each interval to a single vertex.

► Corollary 7. *Let w, k , and m be non-negative integers. Let \prec be a linear ordering of vertices of a graph H , and let $\mathcal{I} = \{L_i : i = 0, 1, \dots\}$ be a partition of $V(H)$ into consecutive intervals in this ordering, where for every $i < j$, $u \in L_i$, and $v \in L_j$, we have $u \succ v$ (note*

the reverse ordering of the indices). Suppose that for each $i \geq 0$, we have $|L_i| \leq m$ and there are at most w indices $j > i$ such that a vertex of L_j has a neighbor in $L_0 \cup L_1 \cup \dots \cup L_i$. Then $|\text{decr}_{H, \prec, k}(v)| \leq m \binom{k+w}{w}$ for each $v \in V(H)$.

Theorem 3 now follows from Lemma 5 and the following theorem.

► **Theorem 8.** Let $f: \mathbb{Z}_0^+ \rightarrow \mathbb{Z}^+$ be a function and let a and e be positive integers. For a graph G and a function $r: V(G) \rightarrow \mathbb{R}^+$, let \prec be a linear ordering of $V(G)$ such that if $u \prec v$, then $r(u) \geq r(v)$. If (G, r) has the property $P(f, a, e)$, then

$$\text{wcol}_{\prec, k}(G) \leq \max(1, \lceil \log_2 k \rceil) f(2k - 2) \binom{k + e + 2}{e + 2}$$

for every integer $k \geq a$.

Proof. Consider any integer $k \geq a$ and a vertex $v \in V(G)$; we are going to bound the number of vertices weakly k -reachable from v . Note that for $k = 1$, $\text{wreach}_{G, \prec, 1}(v)$ consists of the vertices $x \in V(G)$ such that $r(x) \geq r(v)$ and $\lambda_r(v, x) = 0$, and thus $|\text{wreach}_{G, \prec, 1}| \leq f(0)$ by the part (i) of the property $P(f, a, e)$ with $s = 1$ and $p = 0$. Hence, we can assume that $k \geq 2$.

Let H be the graph with the vertex set $\text{wreach}_{G, \prec, k}(v)$, such that for $x, y \in V(H)$ with $x \prec y$, we have $xy \in E(H)$ if and only if there exists a path Q of length at most k in G from v to x such that $y \in V(Q)$ and all the internal vertices of the subpath of Q between x and y are greater than y . Let $\ell(xy)$ denote the minimum length of the subpath between x and y over all paths Q satisfying these conditions. Observe that, by the definition of $V(H)$ and $\ell(xy)$, for every edge e' of H , there exists a decreasing path D from v in H containing the edge e' such that $\sum_{e \in E(D)} \ell(e) \leq k$. Moreover, $V(H) = \text{decr}_{H, \prec, k}(v)$.

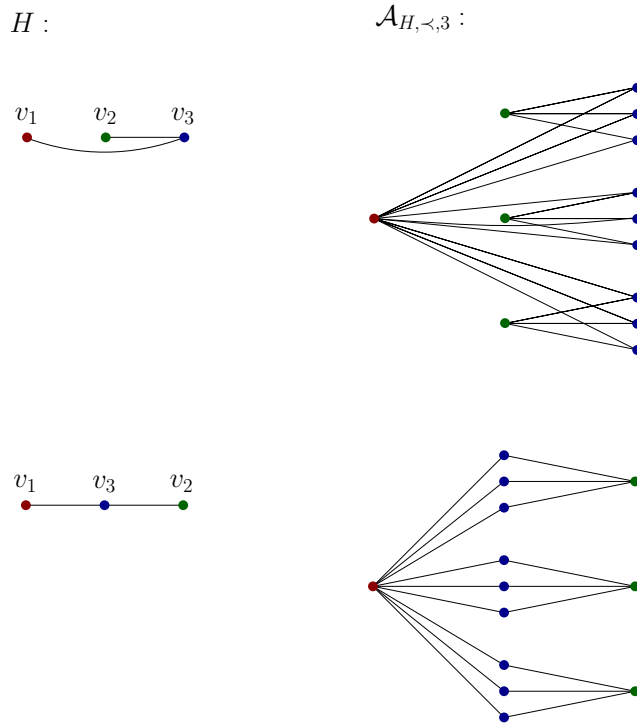
For $i \geq 0$, let L_i consist of the vertices $x \in V(H)$ such that $k^i r(v) \leq r(x) < k^{i+1} r(v)$; in particular, $v \in L_0$. Let $c = \lceil \log_2 k \rceil$ and further partition L_i into $L_{i,1}, \dots, L_{i,c}$, where $L_{i,b}$ consists of the vertices $x \in L_i$ with $2^{b-1} k^i r(v) \leq r(x) < 2^b k^i r(v)$ for $b = 1, \dots, c$. Consider any vertex $x \in L_{i,b}$. Since x is weakly k -reachable from v and $r(x) < 2^b k^i r(v)$, we have $\lambda_r(v, x) < (k - 1) 2^b k^i r(v)$. Moreover, $r(x) \geq 2^{b-1} k^i r(v)$, and thus by the part (i) of the property $P(f, a, e)$ with $s = 2^{b-1} k^i$ and $p = 2(k - 1)$, we conclude $|L_{i,b}| \leq f(2k - 2)$ for each $b \in \{1, \dots, c\}$. Hence, we have $|L_i| = |L_{i,1}| + \dots + |L_{i,c}| \leq cf(2k - 2) = \lceil \log_2 k \rceil f(2k - 2)$.

Let $j_{-1} < j_0 < j_1 < \dots < j_{w-2}$ be all indices such that $j_{-1} > i$ and for each $m \in \{-1, \dots, w - 2\}$, a vertex $u_m \in L_{j_m, m}$ has a neighbor $y_m \in L_0 \cup \dots \cup L_i$ for each m . For $m = 1, \dots, w - 2$, since there exists a decreasing path D from v containing the edge $u_m y_m$ such that $\sum_{e \in E(D)} \ell(e) \leq k$, there exists a path Q in G from v to u_m of length at most k such that $r(x) \leq r(y_m) < k^{i+1} r(v)$ for every internal vertex x of Q . Consequently, we have $\lambda_r(v, u_m) \leq (k - 1) k^{i+1} r(v) \leq s r(v)$ for $s = k^{i+2}$. Moreover, note that $j_m \geq i + 2 + m$, and thus $r(u_m) \geq k^{i+2+m} r(v) \geq a^m s r(v)$. By part (ii) of the property $P(f, a, e)$, we conclude that $w \leq e + 2$.

Hence, Corollary 7 implies that

$$|\text{wreach}_{G, \prec, k}(v)| = |\text{decr}_{H, \prec, k}(v)| \leq \lceil \log_2 k \rceil f(2k - 2) \binom{k + e + 2}{e + 2}$$

for each $v \in V(G)$. ◀



■ **Figure 2** The graph $\mathcal{A}_{H, \prec, 3}$ depicted in two ways, the first respecting the ordering and the second is easier to translate into a geometric setting.

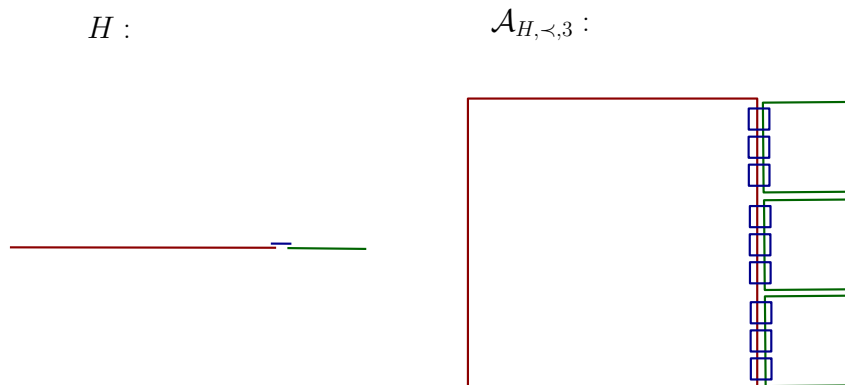
4 Lower bounds

It is relatively easy to construct intersection graphs with large weak coloring numbers with respect to a fixed ordering. The following construction (illustrated in Figure 2) enables us to turn such graphs into graphs that have large weak coloring numbers with respect to every ordering. Let H be a graph and \prec a linear ordering of its vertices. Let $v_1 \prec \dots \prec v_n$ be the vertices of H . Let m be a positive integer and let T be the complete rooted m -ary tree of depth $n - 1$. For $i \in \{1, \dots, n\}$, let $T(v_i)$ be the set of vertices of T at distance exactly $i - 1$ from the root. The graph $\mathcal{A}_{H, \prec, m}$ has vertex set $V(T)$, with vertices $x \in T(v_i)$ and $y \in T(v_j)$ adjacent if and only if $i \neq j$, $v_i v_j \in E(H)$, and x is an ancestor of y in T or vice versa. We say that T is the *scaffolding* of $\mathcal{A}_{H, \prec, m}$.

► **Lemma 9.** *Let k and m be positive integers. Let H be a graph and \prec a linear ordering of its vertices. Suppose that for each $v \in V(H)$, the graph $H[\{u \in V(H) : v \preceq u\}]$ is connected and has diameter at most k . Then*

$$\text{wcol}_k(\mathcal{A}_{H, \prec, m}) \geq \min(m, \text{wcol}_{\prec, k}(H)).$$

Proof. Consider any linear ordering \triangleleft of the vertices of $\mathcal{A}_{H, \prec, m}$. Let T be the scaffolding of $\mathcal{A}_{H, \prec, m}$ and suppose first that there exists a non-leaf vertex $z \in V(T)$ such that all children z_1, \dots, z_m of z in T are smaller than z in the ordering \triangleleft . For $i = 1, \dots, m$, let A_i be the subgraph of $\mathcal{A}_{H, \prec, m}$ induced by z, z_i , and all descendants of z_i in T . Let v be the vertex of



■ **Figure 3** Representation of the graphs H and $\mathcal{A}_{H, \prec, 3}$ in Figure 2 as intersection graphs of intervals and squares.

H such that $z \in T(v)$; since the graph $H[\{u \in V(H) : v \preceq u\}]$ has diameter at most k , every vertex of A_i is at distance at most k from z . Since $z_i \triangleleft z$, we conclude that a vertex of A_i distinct from z is weakly k -reachable from z . Since this is the case for each $i \in \{1, \dots, m\}$ and the subgraphs A_1, \dots, A_m intersect only in z , it follows that

$$\text{wcol}_{\triangleleft, k}(\mathcal{A}_{H, \prec, m}) \geq |\text{wreach}_{\mathcal{A}_{H, \prec, m}, \triangleleft, k}(z)| \geq m.$$

Hence, we can assume that each non-leaf vertex z of T has a child which is greater than z in the ordering \triangleleft . Consequently, T contains a path $u_1 u_2 \dots u_n$ from the root to a leaf such that $u_1 \triangleleft \dots \triangleleft u_n$. The subgraph A of $\mathcal{A}_{H, \prec, m}$ induced by $\{u_1, \dots, u_n\}$ with ordering \triangleleft is isomorphic to H with ordering \prec , and thus

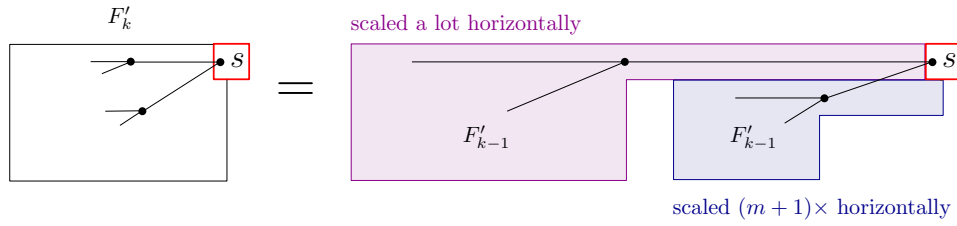
$$\text{wcol}_{\triangleleft, k}(\mathcal{A}_{H, \prec, m}) \geq \text{wcol}_{\triangleleft, k}(A) = \text{wcol}_{\prec, k}(H). \quad \blacktriangleleft$$

Moreover, assuming H has a sufficiently generic representation by comparable axis-aligned boxes, we can also find such a representation for $\mathcal{A}_{H, \prec, m}$. Given an axis-aligned box v in \mathbb{R}^d and $i \in \{1, \dots, d\}$, let $\ell_i(v)$ denote the length of v in the i -th coordinate. We say that a sequence v_1, \dots, v_n of axis-aligned boxes is m -shrinking if $\ell_d(v_i) > m\ell_d(v_{i+1})$ holds for $1 \leq i \leq n - 1$. See Figure 3 for an illustration of the following construction.

► **Lemma 10.** *Let d, t and m be positive integers. Let S be a t -thin finite set of comparable axis-aligned boxes in \mathbb{R}^d and let H be the intersection graph of S . Let T be the scaffolding of $\mathcal{A}_{H, \prec, m}$. Let \prec be a sizewise linear ordering of S and let v_1, \dots, v_n be the sequence of vertices of H in this order. If this sequence is m -shrinking, then $\mathcal{A}_{H, \prec, m}$ is the intersection graph of a t -thin set of comparable axis-aligned boxes in \mathbb{R}^{d+1} , where for $v \in V(H)$ and $u \in T(v)$, u is the product of v with an interval of length $\ell_d(v)$.*

Proof. Let $\varepsilon > 0$ be small enough so that $\ell_d(v_i) \geq m(\ell_d(v_{i+1}) + \varepsilon)$ holds for $1 \leq i \leq n - 1$. For each non-leaf vertex z of T , assign labels $0, \dots, m - 1$ to the edges from z to the children of z in any order; let $l(e)$ denote the label assigned to the edge e . For a vertex y of T , if $y_1 y_2 \dots y_c$ is the path in T from the root to y , then let $l(y) = (l(y_1 y_2), l(y_2 y_3), \dots, l(y_{c-1} y_c))$. Note that y is an ancestor of a vertex x in T if and only if $l(y)$ is a prefix of $l(x)$. Let $s(y) = \sum_{i=1}^{c-1} (l(y))_i (\ell_d(v_{i+1}) + \varepsilon)$, and let $I(y)$ be the interval $[s(y), s(y) + \ell_d(v_c)]$. Observe that if y is an ancestor of a vertex x in T , then $I(x) \subset I(y)$, and if x is neither an ancestor nor a descendant of y in T , then $I(x) \cap I(y) = \emptyset$.

39:12 Weak Coloring Numbers of Intersection Graphs



■ **Figure 4** The construction from Lemma 12.

Hence, letting each vertex y at distance $c - 1$ from the root of T be represented by the box $v_c \times I(y)$ in \mathbb{R}^{d+1} , we obtain a t -thin intersection representation of $\mathcal{A}_{H, \prec, m}$ as described in the statement of the lemma. ◀

To verify the assumptions of Lemma 9, the following concept is useful. Let \prec be a linear ordering of vertices of a graph G . A *decreasing spanning tree* is a spanning tree T of G rooted in the maximum vertex such that any path in T starting in the root is decreasing.

► **Lemma 11.** *Let $k \geq 0$ be an integer. Let \prec be a linear ordering of vertices of a graph G . If G has a decreasing spanning tree T of depth at most k , then $\text{wcol}_{\prec, k}(G) = |V(G)|$, and for each $v \in V(G)$, the graph $G[\{u \in V(H) : v \preceq u\}]$ is connected and has diameter at most $2k$.*

Proof. Let z be the maximum vertex of G . Since T is decreasing and has depth at most k , we have $\text{wreach}_{G, \prec, k}(z) = |V(G)|$. Moreover, for each $v \in V(G)$, letting $C_v = \{u \in V(H) : v \preceq u\}$, observe that for each $x \in C_v$, all ancestors of x also belong to C_v . Hence, $T[C_v]$ is a spanning tree of $G[C_v]$ of depth at most k , and thus $G[C_v]$ is connected and has diameter at most $2k$. ◀

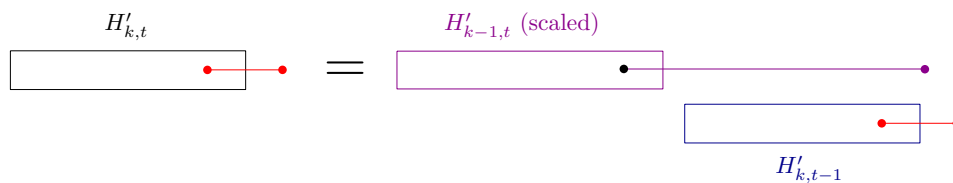
We now find some basic graphs to which we can apply the construction.

► **Lemma 12.** *For all integers $k \geq 0$ and $m \geq 1$, there exists a graph F'_k with $2^{k+1} - 1$ vertices represented as the touching graph of an m -shrinking sequence of comparable axis-aligned rectangles in \mathbb{R}^2 , such that F'_k has a spanning tree of depth at most k decreasing in the sizewise ordering.*

Proof. We proceed by induction on k . For each k , we construct a representation of F'_k where the last vertex is represented by a unit square s and the rest of the representation is contained in the lower left quadrant starting from the middle of the upper side of s . The second coordinate (relevant for the definition of an m -shrinking sequence) is the horizontal one. In the vertical coordinate, all rectangles have length 1. See Figure 4 for an illustration of the construction.

The graph F'_0 is a single vertex represented by s . For $k \geq 1$, to obtain a representation of F'_k , we scale the representation of F'_{k-1} in the horizontal direction by the factor of $m + 1$ and place it so that its upper right corner is the middle of the lower side of s . Then we add another copy of a representation of F'_{k-1} , scaled in the horizontal direction so that all its rectangles are more than m times longer than the already placed ones and so that when we place its upper right corner at the upper left corner of s , their interiors are disjoint from the already placed rectangles.

Observe that F'_k contains a spanning complete binary tree of depth k rooted in s , with the vertices along each path from the root increasing in size, and thus decreasing in the sizewise ordering. ◀



■ **Figure 5** The construction from Lemma 13.

► **Lemma 13.** *For all integers $k \geq 0$ and $m, t \geq 1$, there exists a graph $H'_{k,t}$ with $\binom{k+t}{t}$ vertices represented by a t -thin m -shrinking sequence of intervals in \mathbb{R} , such that $H'_{k,t}$ has a spanning tree of depth at most k decreasing in the sizewise ordering. Furthermore, $H'_{k,t}$ is properly $(t + 1)$ -colorable.*

Proof. We construct a representation of $H'_{k,t}$ with the additional property that the right end of the smallest interval is the strictly rightmost point of the whole representation. See Figure 5 for an illustration of the construction.

We proceed by the induction on $k + t$. If $k = 0$, the representation of $H'_{k,t}$ consists of a single unit interval. If $t = 1$, then the representation consists of an m -shrinking sequence of $k + 1$ intervals intersecting only in endpoints. Hence, suppose that $k \geq 1$ and $t \geq 2$. Then the representation consists of the representation A of $H'_{k,t-1}$ and of the representation B of $H'_{k-1,t}$ scaled so that all its intervals are more than m times longer than all intervals in A and so that when we place the rightmost point of B slightly to the left of the rightmost point of A , only the smallest interval of B intersects all intervals of A .

Observe that $H'_{k,t}$ has a spanning tree of depth k rooted in the smallest vertex, with the vertices along each path from the root increasing in size, and thus decreasing in the sizewise ordering. Finally, note that $H'_{k,t}$ is an interval graph with clique number at most $t + 1$. Since interval graphs are perfect, $H'_{k,t}$ is properly $(t + 1)$ -colorable. ◀

As a final ingredient, we note that we can trade thinness for dimension.

► **Lemma 14.** *For a positive integer d , let $S = \{v_1, \dots, v_n\}$ be a finite set of hypercubes in \mathbb{R}^d , and let G be the intersection graph of S . For any set $Y \subseteq \{1, \dots, n\}$, there exists a set $\{u_1, \dots, u_n\}$ of hypercubes in \mathbb{R}^{d+1} whose intersection graph is isomorphic to G via the isomorphism mapping u_i to v_i for each i , such that*

- for $1 \leq i < j \leq n$, if v_i and v_j have disjoint interiors, then u_i and u_j have disjoint interiors, and
- for $i \in Y$ and $j \in \{1, \dots, n\} \setminus Y$, the hypercubes u_i and u_j have disjoint interiors.

Proof. For $i \in Y$, we set $u_i = v_i \times [0, \ell_1(v_i)]$. For $i \in \{1, \dots, n\} \setminus Y$, we set $u_i = v_i \times [0, -\ell_1(v_i)]$. Note that the intersection of the representation with the hyperplane defined by the last coordinate being 0 is equal to S , and thus indeed the intersection graph of S' is isomorphic to G as described. ◀

► **Corollary 15.** *Let $c \geq 0$ and $d \geq 1$ be integers. If G is a graph of chromatic number at most 2^c representable as an intersection graph of hypercubes in \mathbb{R}^d , then G is also representable as a touching graph of hypercubes in \mathbb{R}^{d+c} .*

Proof. Let $V(G) = \{v_1, \dots, v_n\}$, and let $\varphi: V(G) \rightarrow \{0, 1\}^c$ be a proper coloring of G . By repeatedly applying Lemma 14 for sets Y_1, \dots, Y_c , where $Y_b = \{i \in \{1, \dots, n\} : \varphi(v_i)_b = 0\}$ for $b \in \{1, \dots, c\}$, we obtain a representation of G as an intersection graph of hypercubes u_1, \dots, u_n in \mathbb{R}^{d+c} with the property that for $1 \leq i < j \leq n$, if $\varphi(v_i) \neq \varphi(v_j)$, then u_i

and u_j have disjoint interiors. If $\varphi(v_i) = \varphi(v_j)$, then since φ is a proper coloring, we have $v_i v_j \notin E(G)$, and thus the hypercubes u_i and u_j are disjoint. Consequently, the hypercubes u_1, \dots, u_n have pairwise disjoint interiors. \blacktriangleleft

We are now ready to give the lower bounds.

Proof of Theorem 4. We prove each point separately:

- (i) Let F'_k be the graph obtained in Lemma 12, represented as a touching graph of an m -shrinking sequence of axis-aligned rectangles for $m = 2^{k+1} - 1$. Let \prec be the sizewise ordering of F'_k . By Lemma 11, we have $\text{wcol}_{\prec, k}(F'_k) = |V(F'_k)| = 2^{k+1} - 1$. Letting $F_k = \mathcal{A}_{F'_k, \prec, m}$, Lemma 9 implies $\text{wcol}_{2k}(F_k) \geq 2^{k+1} - 1$. Moreover, by Lemma 10, F_k is a touching graph of comparable axis-aligned boxes in \mathbb{R}^3 .
- (ii) Let $H'_{k,t}$ be the graph obtained in Lemma 13, represented as the intersection graph of a t -thin m -shrinking sequence of intervals for $m = \binom{k+t}{t}$. Let \prec be the sizewise ordering of $H'_{k,t}$. By Lemma 11, we have $\text{wcol}_{\prec, k}(H'_{k,t}) = |V(H'_{k,t})| = \binom{k+t}{t}$. Letting $H_{k,t} = \mathcal{A}_{H'_{k,t}, \prec, m}$, Lemma 9 implies $\text{wcol}_{2k}(H_{k,t}) \geq \binom{k+t}{t}$. Moreover, by Lemma 10, $H_{k,t}$ is the intersection graph of a t -thin set of axis-aligned squares in \mathbb{R}^2 .
- (iii) Recall that by Lemma 13, the graph $H'_{k, 2^d - 1}$ is properly 2^d -colorable. Let T be the scaffolding of $H_{k, 2^d - 1}$. For each $v \in V(H'_{k, 2^d - 1})$, we can assign the color of v to all vertices in $T(v)$, obtaining a proper coloring of $H_{k, 2^d - 1}$ by 2^d colors. Corollary 15 implies that $H_{k, 2^d - 1}$ can be represented as a touching graph of axis-aligned hypercubes in \mathbb{R}^{d+2} . \blacktriangleleft

5 Conclusions

In this paper we have provided upper bounds on the weak coloring number of t -thin intersection graphs of d -dimensional objects of different kinds. Our bounds are qualitatively tight in several aspects. We would like to mention a few open questions, beyond improving the proven upper and lower bounds:

- What is the asymptotic behavior of the k -th weak coloring numbers of planar graphs? It is known to be $O(k^3)$ [14] and $\Omega(k^2 \log k)$ [9].
- What is the asymptotic behavior of the k -th strong coloring numbers of touching graphs of unit balls in \mathbb{R}^d ? It is known to be $O(k^{d-1})$ and $\Omega(k^{d/2})$.

References

- 1 Vida Dujmović, Pat Morin, and David R. Wood. Graph product structure for non-minor-closed classes. *arXiv*, 1907.05168, 2019. [arXiv:1907.05168](#).
- 2 Zdeněk Dvořák. Constant-factor approximation of domination number in sparse graphs. *European Journal of Combinatorics*, 34:833–840, 2013.
- 3 Zdeněk Dvořák, Rose McCarty, and Sergey Norin. Sublinear separators in intersection graphs of convex shapes. *SIAM Journal on Discrete Mathematics*, 35(2):1149–1164, 2021. [doi:10.1137/20M1311156](#).
- 4 Zdeněk Dvořák, Daniel Gonçalves, Abhiruk Lahiri, Jane Tan, and Torsten Ueckerdt. On comparable box dimension. Manuscript.
- 5 Kord Eickmeyer, Archontia C. Giannopoulou, Stephan Kreutzer, O-joung Kwon, Michał Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. Neighborhood complexity and kernelization for nowhere dense classes of graphs. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63:1–63:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. [doi:10.4230/LIPIcs.ICALP.2017.63](#).

- 6 Louis Esperet and Jean-Florent Raymond. Polynomial expansion and sublinear separators. *European Journal of Combinatorics*, 69:49–53, 2018.
- 7 Martin Grohe, Stephan Kreutzer, Roman Rabinovich, Sebastian Siebertz, and Konstantinos Stavropoulos. Coloring and covering nowhere dense graphs. *SIAM Journal on Discrete Mathematics*, 32:2467–2481, 2018.
- 8 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 89–98. ACM, 2014.
- 9 Gwenaël Joret and Piotr Micek. Improved bounds for weak coloring numbers. *CoRR*, 2021. [arXiv:2102.10061](https://arxiv.org/abs/2102.10061).
- 10 Hal A. Kierstead and Daqing Yang. Orderings on graphs and game coloring number. *Order*, 20(3):255–264, 2003.
- 11 Paul Koebe. Kontaktprobleme der Konformen Abbildung. *Math.-Phys. Kl.*, 88:141–164, 1936.
- 12 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity (Graphs, Structures, and Algorithms)*, volume 28 of *Algorithms and Combinatorics*. Springer, 2012.
- 13 Felix Reidl and Blair D. Sullivan. A color-avoiding approach to subgraph counting in bounded expansion classes. *arXiv*, 2001.05236, 2020. [arXiv:2001.05236](https://arxiv.org/abs/2001.05236).
- 14 Jan van den Heuvel, Patrice Ossona de Mendez, Daniel Quiroz, Roman Rabinovich, and Sebastian Siebertz. On the generalised colouring numbers of graphs that exclude a fixed minor. *European Journal of Combinatorics*, 66:129–144, 2017.
- 15 Xuding Zhu. Colouring graphs with bounded generalized colouring number. *Discrete Math.*, 309(18):5562–5568, 2009.

ε -Isometric Dimension Reduction for Incompressible Subsets of ℓ_p

Alexandros Eskenazis   

Trinity College and Department of Pure Mathematics and Mathematical Statistics,
University of Cambridge, UK

Abstract

Fix $p \in [1, \infty)$, $K \in (0, \infty)$ and a probability measure μ . We prove that for every $n \in \mathbb{N}$, $\varepsilon \in (0, 1)$ and $x_1, \dots, x_n \in L_p(\mu)$ with $\left\| \max_{i \in \{1, \dots, n\}} |x_i| \right\|_{L_p(\mu)} \leq K$, there exists $d \leq \frac{32e^2(2K)^{2p} \log n}{\varepsilon^2}$ and vectors $y_1, \dots, y_n \in \ell_p^d$ such that

$$\forall i, j \in \{1, \dots, n\}, \quad \|x_i - x_j\|_{L_p(\mu)}^p - \varepsilon \leq \|y_i - y_j\|_{\ell_p^d}^p \leq \|x_i - x_j\|_{L_p(\mu)}^p + \varepsilon.$$

Moreover, the argument implies the existence of a greedy algorithm which outputs $\{y_i\}_{i=1}^n$ after receiving $\{x_i\}_{i=1}^n$ as input. The proof relies on a derandomized version of Maurey's empirical method (1981) combined with a combinatorial idea of Ball (1990) and a suitable change of measure. Motivated by the above embedding, we introduce the notion of ε -isometric dimension reduction of the unit ball \mathbf{B}_E of a normed space $(E, \|\cdot\|_E)$ and we prove that \mathbf{B}_{ℓ_p} does not admit ε -isometric dimension reduction by linear operators for any value of $p \neq 2$.

2012 ACM Subject Classification Theory of computation \rightarrow Random projections and metric embeddings; Mathematics of computing \rightarrow Probabilistic algorithms; Mathematics of computing \rightarrow Approximation

Keywords and phrases Dimension reduction, ε -isometric embedding, Maurey's empirical method, change of measure

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.40

Funding The author was supported by a Junior Research Fellowship from Trinity College, Cambridge.

Acknowledgements I am grateful to Keith Ball, Assaf Naor and Pierre Youssef for insightful discussions and useful feedback. I also wish to thank the anonymous referees for their constructive comments.

1 Introduction

1.1 Metric dimension reduction

Using standard terminology from metric embeddings (see [35]), we say that a mapping between metric spaces $f : (M, d_M) \rightarrow (N, d_N)$ is a bi-Lipschitz embedding with distortion at most $\alpha \in [1, \infty)$ if there exists a scaling factor $\sigma \in (0, \infty)$ such that

$$\forall x, y \in M, \quad \sigma d_M(x, y) \leq d_N(f(x), f(y)) \leq \alpha \sigma d_M(x, y). \quad (1)$$

Throughout this paper, we shall denote by ℓ_p^d the linear space \mathbb{R}^d equipped with the p -norm,

$$\forall a = (a_1, \dots, a_d) \in \mathbb{R}^d, \quad \|a\|_{\ell_p^d} = \left(\sum_{i=1}^d |a_i|^p \right)^{1/p}. \quad (2)$$

The classical Johnson–Lindenstrauss lemma [20] asserts that if $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$ is a Hilbert space and $x_1, \dots, x_n \in \mathcal{H}$, then for every $\varepsilon \in (0, 1)$ there exist $d \leq \frac{C \log n}{\varepsilon^2}$ and $y_1, \dots, y_n \in \ell_2^d$ such that

$$\forall i, j \in \{1, \dots, n\}, \quad \|x_i - x_j\|_{\mathcal{H}} \leq \|y_i - y_j\|_{\ell_2^d} \leq (1 + \varepsilon) \cdot \|x_i - x_j\|_{\mathcal{H}}, \quad (3)$$



© Alexandros Eskenazis;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 40; pp. 40:1–40:14
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



where $C \in (0, \infty)$ is a universal constant. In the above embedding terminology, the Johnson–Lindenstrauss lemma states that for every $\varepsilon \in (0, 1)$, $n \in \mathbb{N}$ and $d \geq \frac{C \log n}{\varepsilon^2}$, any n -point subset of Hilbert space admits a bi-Lipschitz embedding into ℓ_2^d with distortion at most $1 + \varepsilon$. In order to prove their result, Johnson and Lindenstrauss introduced in [20] the influential random projection method that has since had many important applications in metric geometry and theoretical computer science and kickstarted the field of *metric dimension reduction* (see the recent survey [33] of Naor) which lies at the intersection of those two subjects.

Following [33], we say that an infinite dimensional Banach space $(E, \|\cdot\|_E)$ admits bi-Lipschitz dimension reduction if there exists $\alpha = \alpha(E) \in [1, \infty)$ such that for every $n \in \mathbb{N}$, there exists $k_n = k_n(E, \alpha) \in \mathbb{N}$ satisfying

$$\lim_{n \rightarrow \infty} \frac{\log k_n}{\log n} = 0 \tag{4}$$

and such that any n -point subset \mathcal{S} of E admits a bi-Lipschitz embedding with distortion at most α in a finite-dimensional linear subspace F of E with $\dim F \leq k_n$. The only non-Hilbertian space that is known to admit bi-Lipschitz dimension reduction is the 2-convexification of the classical Tsirelson space, as proven by Johnson and Naor in [21]. Turning to negative results, Matoušek proved in [30] the impossibility of bi-Lipschitz dimension reduction in ℓ_∞ , whereas Brinkman and Charikar [10] (see also [28] for a shorter proof) constructed an n -point subset of ℓ_1 which does not admit a bi-Lipschitz embedding into any $n^{o(1)}$ -dimensional subspace of ℓ_1 . Their theorem was recently refined by Naor, Pisier and Schechtman [34] who showed that the same n -point subset of ℓ_1 does not embed into any $n^{o(1)}$ -dimensional subspace of the trace class S_1 (see also the striking recent work [38] of Regev and Vidick, where the impossibility of polynomial almost isometric dimension reduction in S_1 is established). We refer to [33, Theorem 16] for a summary of the best known bounds quantifying the aforementioned qualitative statements. Despite the lapse of almost four decades since the proof of the Johnson–Lindenstrauss lemma, the following natural question remains stubbornly open.

► **Question 1.** *For which values of $p \notin \{1, 2, \infty\}$ does ℓ_p admit bi-Lipschitz dimension reduction?*

1.2 Dimensionality and structure

An important feature of the formalism of bi-Lipschitz dimension reduction in a Banach space E is that both the distortion $\alpha(E)$ of the embedding and the dimension $k_n(E, \alpha)$ of the target subspace F are independent of the given n -point subset \mathcal{S} of E . Nevertheless, there are instances in which one can construct delicate embeddings whose distortion or the dimension of their targets depends on subtle geometric parameters of \mathcal{S} . For instance, we mention an important theorem of Schechtman [39, Theorem 5] (which built on work of Klartag and Mendelson [24]) who constructed a linear embedding of an arbitrary subset \mathcal{S} of ℓ_2 into any Banach space E whose distortion depends only on the Gaussian width of \mathcal{S} and the ℓ -norm of the identity operator $\text{id}_E : E \rightarrow E$. In the special case that E is a Hilbert space, a substantially richer family of such embeddings was devised in [29].

Let μ be a probability measure on a measurable space Ω . As usual, we shall denote the $L_p(\mu)$ -norm of a function $f : \Omega \rightarrow \mathbb{R}$ by

$$\|f\|_{L_p(\mu)} \stackrel{\text{def}}{=} \left(\int |f|^p \, d\mu \right)^{1/p}. \tag{5}$$

For a subset \mathcal{S} of $L_p(\mu)$, we shall denote by

$$\mathcal{J}(\mathcal{S}) \stackrel{\text{def}}{=} \left\| \max_{x \in \mathcal{S}} |x| \right\|_{L_p(\mu)} \quad (6)$$

the $L_p(\mu)$ -norm of the pointwise maximum of all functions in \mathcal{S} and we will say that \mathcal{S} is K -incompressible¹ if $\mathcal{J}(\mathcal{S}) \leq K$. The main contribution of the present paper is the following dimensionality reduction theorem for incompressible subsets of $L_p(\mu)$ which, in contrast to all the results discussed earlier, is valid for *any* value of $p \in [1, \infty)$.

► **Theorem 2** (ε -isometric dimension reduction for incompressible subsets of $L_p(\mu)$). *Fix parameters $p \in [1, \infty)$, $n \in \mathbb{N}$, $K \in (0, \infty)$ and let $\{x_i\}_{i=1}^n$ be a K -incompressible family of vectors in $L_p(\mu)$ for some probability measure μ . Then for every $\varepsilon \in (0, 1)$, there exists $d \in \mathbb{N}$ with $d \leq \frac{32e^2(2K)^{2p} \log n}{\varepsilon^2}$ and points $y_1, \dots, y_n \in \ell_p^d$ such that*

$$\forall i, j \in \{1, \dots, n\}, \quad \|x_i - x_j\|_{L_p(\mu)}^p - \varepsilon \leq \|y_i - y_j\|_{\ell_p^d}^p \leq \|x_i - x_j\|_{L_p(\mu)}^p + \varepsilon. \quad (7)$$

Besides the appearance of the incompressibility parameter K in the bound for the dimension d of the target space, Theorem 2 differs from the Johnson–Lindenstrauss lemma in that the error in (7) is *additive* rather than *multiplicative*. Recall that a map between metric spaces $f : (\mathcal{M}, d_m) \rightarrow (\mathcal{N}, d_n)$ is called an ε -isometric embedding if

$$\forall x, y \in \mathcal{M}, \quad |d_n(f(x), f(y)) - d_m(x, y)| \leq \varepsilon. \quad (8)$$

Embeddings with additive errors occur naturally in metric geometry and, more specifically, in metric dimension reduction (see e.g. [42, Section 9.3]). We mention for instance a result [37, Theorem 1.5] of Plan and Vershynin who showed that any subset \mathcal{S} of the unit sphere in ℓ_2^n admits a δ -isometric embedding into the d -dimensional Hamming cube $(\{-1, 1\}^d, \|\cdot\|_1)$, where d depends polynomially on δ^{-1} and the Gaussian width of \mathcal{S} . In the above embedding terminology and in view of the elementary inequality $|\alpha - \beta| \leq |\alpha^p - \beta^p|^{1/p}$ which holds for every $\alpha, \beta > 0$, Theorem 2 asserts that any n -point K -incompressible subset of $L_p(\mu)$ admits an $\varepsilon^{1/p}$ -isometric embedding into ℓ_p^d for the above choice of dimension d . For further occurrences of ε -isometric embeddings in the dimensionality reduction and compressed sensing literatures, we refer to [37, 18, 19, 29, 42, 8] and the references therein.

1.3 Method of proof

A large part of the (vast) literature on metric dimension reduction focuses on showing that a typical low-rank linear operator chosen randomly from a specific ensemble acts as an approximate isometry on a given set \mathcal{S} with high probability. For subsets \mathcal{S} of Euclidean space, this principle has been confirmed for random projections [20, 13, 11, 33], matrices with Gaussian [14, 15, 39], Rademacher [5, 1] and subgaussian [24, 16, 12, 29] entries, randomizations of matrices with the RIP [25] as well as more computationally efficient models [31, 2, 3, 23, 9] which are based on sparse matrices. Beyond its inherent interest as an ℓ_p -dimension reduction theorem (albeit, for specific configurations of points), Theorem 2 also differs from the aforementioned works in its method of proof. The core of the argument, rather than sampling from a random matrix ensemble, relies on Maurey’s empirical method [36] (see

¹ The terminology is borrowed by the standard use of the term “incompressible vector” from random matrix theory, which refers to points on the unit sphere of \mathbb{R}^n which are far from the coordinate vectors e_1, \dots, e_n .

Section 2.1) which is a dimension-free way to approximate points in bounded convex subsets of Banach spaces by convex combinations of extreme points with prescribed length. An application of the method to the positive cone of L_p -distance matrices (the use of which in this context is inspired by classical work of Ball [6]) equipped with the supremum norm allows us to deduce (see Proposition 7) the conclusion of Theorem 2 under the stronger assumption that

$$K \geq \max_{i \in \{1, \dots, n\}} \|x_i\|_{L_\infty(\mu)}. \quad (9)$$

While Maurey’s empirical method is an a priori existential statement that is proven via the probabilistic method, recent works (see [7, 17]) have focused on derandomizing its proof for specific Banach spaces. In the setting of Theorem 2, we can use these tools to show (see Corollary 13) that there exists a greedy algorithm which receives as input the high-dimensional data $\{x_i\}_{i=1}^n$ and produces as output the low-dimensional points $\{y_i\}_{i=1}^n$. Finally, using a suitable change of measure [32] (see Section 2.3) we are able to relax the stronger assumption (9) to that of K -incompressibility and derive the conclusion of Theorem 2. Finally, we emphasize that, in contrast to most of the dimension reduction algorithms (randomized or not) discussed earlier, the one which gives Theorem 2 is not *oblivious* but is rather tailored to the specific configuration of points $\{x_i\}_{i=1}^n$ as it relies on the use of Maurey’s empirical method.

1.4 ε -isometric dimension reduction

Given two moduli $\omega, \Omega : [0, \infty) \rightarrow [0, \infty)$, we say (following [33]) that a Banach space $(E, \|\cdot\|_E)$ admits metric dimension reduction with moduli (ω, Ω) if for any $n \in \mathbb{N}$ there exists $k_n = k_n(E) \in \mathbb{N}$ with $k_n = n^{o(1)}$ as $n \rightarrow \infty$ such that for any $x_1, \dots, x_n \in E$, there exists a subspace F of E with $\dim F \leq k_n$ and $y_1, \dots, y_n \in F$ satisfying

$$\forall i, j \in \{1, \dots, n\}, \quad \omega(\|x_i - x_j\|_E) \leq \|y_i - y_j\|_E \leq \Omega(\|x_i - x_j\|_E). \quad (10)$$

In view of Theorem 2, we would be interested in formulating a suitable notion of dimension reduction via ε -isometric embeddings which would be fitting to the moduli appearing in (7).

► **Remark 3.** Let $a, b \in (0, \infty)$, suppose that $\omega, \Omega : [0, \infty) \rightarrow [0, \infty)$ are two moduli satisfying

$$\lim_{t \rightarrow \infty} \frac{\omega(t)}{t} = a \quad \text{and} \quad \lim_{t \rightarrow \infty} \frac{\Omega(t)}{t} = b \quad (11)$$

and that the Banach space $(E, \|\cdot\|_E)$ admits metric dimension reduction with moduli (ω, Ω) . Fix $n \in \mathbb{N}$ and $x_1, \dots, x_n \in E$. Applying the assumption (10) to the points sx_1, \dots, sx_n where $s \gg 1$, we deduce that there exist points $y_1(s), \dots, y_n(s)$ in a k_n -dimensional subspace $F(s)$ of E such that

$$\forall i, j \in \{1, \dots, n\}, \quad \omega(s\|x_i - x_j\|_E) \leq \|y_i(s) - y_j(s)\|_E \leq \Omega(s\|x_i - x_j\|_E). \quad (12)$$

For any $\eta \in (0, 1)$, we can then choose s large enough (as a function of η and the x_i) such that

$$\forall i, j \in \{1, \dots, n\}, \quad (1 - \eta)a\|x_i - x_j\|_E \leq \frac{\|y_i(s) - y_j(s)\|_E}{s} \leq (1 + \eta)b\|x_i - x_j\|_E. \quad (13)$$

Therefore, we conclude that E also admits bi-Lipschitz dimension reduction (with distortion b/a).

This simple scaling argument suggests that any reasonable notion of ε -isometric dimension reduction can differ from the corresponding bi-Lipschitz theory only in small scales, thus motivating the following definition. We denote by \mathbf{B}_E the closed unit ball of a normed space $(E, \|\cdot\|_E)$, that is $\mathbf{B}_E = \{x \in E : \|x\|_E \leq 1\}$.

► **Definition 4** (ε -isometric dimension reduction). Fix $\varepsilon \in (0, 1)$, $r \in (0, \infty)$ and let $(E, \|\cdot\|_E)$ be an infinite-dimensional Banach space. We say that \mathbf{B}_E admits ε -isometric dimension reduction with power r if for every $n \in \mathbb{N}$ there exists $k_n = k_n^r(E, \varepsilon) \in \mathbb{N}$ with $k_n = n^{o(1)}$ as $n \rightarrow \infty$ for which the following condition holds. For every n points $x_1, \dots, x_n \in \mathbf{B}_E$ there exists a linear subspace F of E with $\dim F \leq k_n$ and points $y_1, \dots, y_n \in F$ satisfying

$$\forall i, j \in \{1, \dots, n\}, \quad \|x_i - x_j\|_E^r - \varepsilon \leq \|y_i - y_j\|_E^r \leq \|x_i - x_j\|_E^r + \varepsilon. \tag{14}$$

The fact that the whole space ℓ_2 admits ε -isometric dimension reduction with $r = 1$ and corresponding target dimension $k_n^1(\ell_2, \varepsilon) \lesssim \frac{\log n}{\varepsilon^2}$ follows from the additive version of the Johnson–Lindenstrauss lemma, first proven by Liaw, Mehrabian, Plan and Vershynin [29] (see also [42, Proposition 9.3.2]). In Corollary 9 we obtain the same conclusion for its unit ball \mathbf{B}_{ℓ_2} with a slightly weaker bound for the target dimension using our Theorem 2.

It is clear from the definitions that if a Banach space E admits bi-Lipschitz dimension reduction with distortion $\frac{1+\varepsilon}{1-\varepsilon}$, where $\varepsilon \in (0, 1)$, then \mathbf{B}_E admits 2ε -isometric dimension reduction with power $r = 1$. The ε -isometric analogue of Question 1 deserves further investigation.

► **Question 5.** For which values of $p \neq 2$ does \mathbf{B}_{ℓ_p} admit ε -isometric dimension reduction?

Even though the K -incompressibility assumption of Theorem 2 may a priori seem restrictive, it is satisfied for *most* configurations of points in \mathbf{B}_{ℓ_p} . Suppose that $n, N \in \mathbb{N}$ such that N is polynomial² in n . Then, standard considerations (see Remark 10) show that with high probability, a uniformly chosen n -point subset S of $N^{1/p} \mathbf{B}_{\ell_p^N}$ is $O(\log n)^{1/p}$ -incompressible.

1.5 ε -isometric dimension reduction by linear maps

A close inspection of the proof of Theorem 2 (see Remark 12) reveals that in fact the low-dimensional points $\{y_i\}_{i=1}^n$ can be realized as images of the initial data $\{x_i\}_{i=1}^n$ under a carefully chosen linear operator. Nevertheless, we will show that for any $p \neq 2$ and n large enough, there exist an n -point subset of \mathbf{B}_{ℓ_p} whose image under any fixed linear ε -isometric embedding has rank which is linear in n . In fact, we shall prove the following more general statement which refines a theorem that Lee, Mendel and Naor proved in [27] for bi-Lipschitz embeddings.

► **Theorem 6** (Impossibility of linear dimension reduction in \mathbf{B}_{ℓ_p}). Fix $p \neq 2$ and two moduli $\omega, \Omega : [0, \infty) \rightarrow [0, \infty)$ with $\omega(1) > 0$. For arbitrarily large $n \in \mathbb{N}$, there exists an n -point subset $\mathcal{S}_{n,p}$ of \mathbf{B}_{ℓ_p} such that the following holds. If $T : \text{span}(\mathcal{S}_{n,p}) \rightarrow \ell_p^d$ is a linear operator satisfying

$$\forall x, y \in \mathcal{S}_{n,p}, \quad \omega(\|x - y\|_{\ell_p}) \leq \|Tx - Ty\|_{\ell_p^d} \leq \Omega(\|x - y\|_{\ell_p}), \tag{15}$$

then $d \geq \left(\frac{\omega(1)}{\Omega(1)}\right)^{\frac{2p}{|p-2|}} \cdot \frac{n-1}{2}$.

² This relation between the parameters n, N is natural as any n -point subset of ℓ_p embeds isometrically in ℓ_p^N with $N = \binom{n}{2} + 1$ by Ball’s isometric embedding theorem [6].

2 Proof of Theorem 2

We say that a normed space $(E, \|\cdot\|_E)$ has Rademacher type p if there exists a universal constant $T \in (0, \infty)$ such that for every $n \in \mathbb{N}$ and every $x_1, \dots, x_n \in E$,

$$\frac{1}{2^n} \sum_{\varepsilon \in \{-1, 1\}^n} \left\| \sum_{i=1}^n \varepsilon_i x_i \right\|_E^p \leq T^p \sum_{i=1}^n \|x_i\|_E^p. \quad (16)$$

The least constant T such that (16) is satisfied is denoted by $T_p(E)$. A standard symmetrization argument (see [26, Proposition 9.11]) shows that if X_1, \dots, X_n are independent E -valued random variables with $\mathbb{E}[X_i] = 0$ for every $i \in \{1, \dots, n\}$, then

$$\mathbb{E} \left\| \sum_{i=1}^n X_i \right\|_E^p \leq (2T_p(E))^p \sum_{i=1}^n \mathbb{E} \|X_i\|_E^p. \quad (17)$$

2.1 Maurey's empirical method and its algorithmic counterparts

A classical theorem of Carathéodory asserts that if \mathcal{T} is a subset of \mathbb{R}^m , then any point $z \in \text{conv}(\mathcal{T})$ can be expressed as a convex combination of at most $m+1$ points of \mathcal{T} . Maurey's empirical method is a powerful dimension-free approximate version of Carathéodory's theorem, first popularized in [36], that has numerous applications in geometry and theoretical computer science. Let $(E, \|\cdot\|_E)$ be a Banach space, consider a bounded subset \mathcal{T} of E and fix $z \in \text{conv}(\mathcal{T})$. Since z is a convex combination of elements of \mathcal{T} , there exists $m \in \mathbb{N}$, $\lambda_1, \dots, \lambda_m \in (0, \infty)$ and $t_1, \dots, t_m \in \mathcal{T}$ such that

$$\sum_{k=1}^m \lambda_k = 1 \quad \text{and} \quad z = \sum_{k=1}^m \lambda_k t_k. \quad (18)$$

Let X be an E -valued discrete random variable with $\mathbb{P}\{X = t_k\} = \lambda_k$ for all $k \in \{1, \dots, m\}$ and consider X_1, \dots, X_d i.i.d. copies of X . Then, conditions (18) ensure that X is well defined and $\mathbb{E}[X] = z$. Therefore, applying the Rademacher type condition (17) to the centered random variables $\{X_s - z\}_{s=1}^d$ and normalizing, we get

$$\mathbb{E} \left\| \frac{1}{d} \sum_{s=1}^d X_s - z \right\|_E^p \leq \frac{(2T_p(E))^p}{d^{p-1}} \mathbb{E} \|X - z\|_E^p. \quad (19)$$

Since X takes values in \mathcal{T} , if $\mathcal{T} \subseteq R\mathbf{B}_E$, we then deduce that there exist $x_1, \dots, x_d \in \mathcal{T}$ such that

$$\left\| \frac{1}{d} \sum_{s=1}^d x_s - z \right\|_E \leq \frac{4RT_p(E)}{d^{1-1/p}}. \quad (20)$$

While the above argument is probabilistic, recent works have focused on derandomizing Maurey's sampling lemma for smaller classes of Banach spaces, thus constructing deterministic algorithms which output the empirical approximation $\frac{x_1 + \dots + x_d}{d}$ of z . The first result in this direction is due to Barman [7] who treated the case that E is an $L_r(\mu)$ -space, $r \in (1, \infty)$. This assumption was recently generalized by Ivanov in [17] who built a greedy algorithm which constructs the desired empirical mean in an arbitrary p -uniformly smooth space.

2.2 Dimension reduction in $L_p(\mu)$ for uniformly bounded vectors

With Maurey’s empirical method at hand, we are ready to proceed to the first part of the proof of Theorem 2, namely the ε -isometric dimension reduction property of $L_p(\mu)$ under the strong assumption that the given point set consists of functions which are bounded in $L_\infty(\mu)$.

► **Proposition 7.** Fix $p \in [1, \infty)$, $n \in \mathbb{N}$ and let $\{x_i\}_{i=1}^n$ be a family of vectors in $L_p(\mu)$ for some probability measure μ . Denote by $L = \max_{i \in \{1, \dots, n\}} \|x_i\|_{L_\infty(\mu)} \in [0, \infty]$. Then for every $\varepsilon \in (0, 1)$, there exists $d \in \mathbb{N}$ with $d \leq \frac{32e^2(2L)^{2p} \log n}{\varepsilon^2}$ and $y_1, \dots, y_n \in \ell_p^d$ such that

$$\forall i, j \in \{1, \dots, n\}, \quad \|x_i - x_j\|_{L_p(\mu)}^p - \varepsilon \leq \|y_i - y_j\|_{\ell_p^d}^p \leq \|x_i - x_j\|_{L_p(\mu)}^p + \varepsilon. \tag{21}$$

Proof. We shall identify $\ell_\infty^{\binom{n}{2}}$ with the vector space of all symmetric $n \times n$ real matrices with 0 on the diagonal equipped with the supremum norm. Consider the set

$$\mathcal{C}_p = \left\{ (\|z_i - z_j\|_{L_p(\rho)}^p)_{i,j=1, \dots, n} : \rho \text{ is a probability measure and } z_1, \dots, z_n \in L_p(\rho) \right\} \subseteq \ell_\infty^{\binom{n}{2}}.$$

It is obvious that \mathcal{C}_p is a cone in the sense that $\mathcal{C}_p = \lambda \mathcal{C}_p$ for every $\lambda > 0$ but moreover \mathcal{C}_p is convex. To see this, consider $A, B \in \mathcal{C}_p$, probability spaces $(\Omega_1, \rho_1), (\Omega_2, \rho_2)$ and vectors $\{z_i\}_{i=1}^n, \{w_i\}_{i=1}^n$ in $L_p(\rho_1)$ and $L_p(\rho_2)$ respectively such that

$$\forall i, j \in \{1, \dots, n\}, \quad A_{ij} = \|z_i - z_j\|_{L_p(\rho_1)}^p \quad \text{and} \quad B_{ij} = \|w_i - w_j\|_{L_p(\rho_2)}^p. \tag{22}$$

Fix $\lambda \in (0, 1)$ and consider the disjoint union $\Omega_1 \sqcup \Omega_2$ of Ω_1 and Ω_2 equipped with the probability measure $\rho(\lambda) = \lambda \rho_1 + (1 - \lambda) \rho_2$. Then, by (22) the functions $\zeta_i : \Omega_1 \sqcup \Omega_2 \rightarrow \mathbb{R}$ given by $\zeta_i|_{\Omega_1} = z_i$ and $\zeta_i|_{\Omega_2} = w_i$, where $i \in \{1, \dots, n\}$, belong in $L_p(\rho(\lambda))$ and for every $i, j \in \{1, \dots, n\}$ satisfy the conditions

$$\|\zeta_i - \zeta_j\|_{L_p(\rho(\lambda))}^p = \lambda \|z_i - z_j\|_{L_p(\rho_1)}^p + (1 - \lambda) \|w_i - w_j\|_{L_p(\rho_2)}^p = \lambda A_{ij} + (1 - \lambda) B_{ij}, \tag{23}$$

which ensure that $\lambda A + (1 - \lambda) B \in \mathcal{C}_p$, making \mathcal{C}_p a convex cone. Consider the embedding $\mathcal{M} : L_p(\mu)^n \rightarrow \mathcal{C}_p$ mapping a vector $z = (z_1, \dots, z_n)$ to the corresponding distance matrix, i.e.

$$\forall i, j \in \{1, \dots, n\}, \quad \mathcal{M}(z)_{ij} = \|z_i - z_j\|_{L_p(\mu)}^p. \tag{24}$$

Without loss of generality we will assume that the given points $x_1, \dots, x_n \in L_p(\mu)$ are simple functions with $\|x_i\|_{L_\infty(\mu)} \leq L$. Let $\{S_1, \dots, S_m\}$ be a partition of the underlying measure space such that each x_i is constant on each S_k and suppose that $x_i|_{S_k} = a(i, k) \in [-L, L]$ for $i \in \{1, \dots, n\}$ and $k \in \{1, \dots, m\}$. Then, for every $i, j \in \{1, \dots, n\}$, we have

$$\mathcal{M}(x)_{ij} = \sum_{k=1}^m \int_{S_k} |x_i - x_j|^p \, d\mu = \sum_{k=1}^m \mu(S_k) \cdot |a(i, k) - a(j, k)|^p = \sum_{k=1}^m \mu(S_k) \mathcal{M}(y(k))_{ij}, \tag{25}$$

where $y(k) \stackrel{\text{def}}{=} (a(1, k), \dots, a(n, k)) \in L_p(\mu)^n$ is a vector whose components are constant functions. As μ is a probability measure and $\{S_1, \dots, S_m\}$ is a partition, identity (25) implies that

$$\mathcal{M}(x) \in \text{conv} \{ \mathcal{M}(y(k)) : k \in \{1, \dots, m\} \} \subseteq \ell_\infty^{\binom{n}{2}}. \tag{26}$$

40:8 Dimension Reduction for Incompressible Subsets of ℓ_p

Observe that since $a(i, k) \in [-L, L]$ for every $i \in \{1, \dots, n\}$ and $k \in \{1, \dots, m\}$, we have

$$\forall k \in \{1, \dots, m\}, \quad \|\mathcal{M}(y(k))\|_{\ell_\infty^{\binom{n}{2}}} = \max_{i, j \in \{1, \dots, n\}} |a(i, k) - a(j, k)|^p \leq (2L)^p. \quad (27)$$

Moreover, $\ell_\infty^{\binom{n}{2}}$ is e -isomorphic to $\ell_{p_n}^{\binom{n}{2}}$ where $p_n = \log \binom{n}{2}$. It is well-known (see [26, Chapter 9]) that $T_2(\ell_p) \leq \sqrt{p-1}$ for every $p \geq 2$ and thus

$$T_2(\ell_\infty^{\binom{n}{2}}) \leq e\sqrt{p_n-1} < \sqrt{2e^2 \log n}. \quad (28)$$

Applying Maurey's sampling lemma (Section 2.1) while taking into account (27) and (28), we deduce that for every $d \geq 1$ there exist $k_1, \dots, k_d \in \{1, \dots, m\}$ such that

$$\left\| \frac{1}{d} \sum_{s=1}^d \mathcal{M}(y(k_s)) - \mathcal{M}(x) \right\|_{\ell_\infty^{\binom{n}{2}}} \leq \frac{2^{p+\frac{5}{2}} e L^p \sqrt{\log n}}{\sqrt{d}}. \quad (29)$$

Therefore, if $\varepsilon \in (0, 1)$ is such that $d \geq \frac{32e^2(2L)^{2p} \log n}{\varepsilon^2}$ we then have

$$\forall i, j \in \{1, \dots, n\}, \quad \left| \frac{1}{d} \sum_{s=1}^d |a(i, k_s) - a(j, k_s)|^p - \|x_i - x_j\|_{L_p(\mu)}^p \right| \leq \varepsilon. \quad (30)$$

Finally, consider for each $i \in \{1, \dots, n\}$ a vector $y_i = (y_i(1), \dots, y_i(d)) \in \ell_p^d$ given by

$$\forall s \in \{1, \dots, d\}, \quad y_i(s) = \frac{a(i, k_s)}{d^{1/p}} \quad (31)$$

and notice that (30) can be equivalently rewritten as

$$\forall i, j \in \{1, \dots, n\}, \quad \|x_i - x_j\|_{L_p(\mu)}^p - \varepsilon \leq \|y_i - y_j\|_{\ell_p^d}^p \leq \|x_i - x_j\|_{L_p(\mu)}^p + \varepsilon, \quad (32)$$

concluding the proof of the proposition. \blacktriangleleft

► Remark 8. Following a comment by an anonymous referee, we point out that the existence of k_1, \dots, k_d satisfying (29) can be proven without relying on the type estimate $T_2(\ell_\infty^m) = O(\sqrt{\log m})$. Indeed, this follows by sampling k_1, \dots, k_d independently from $\{1, \dots, m\}$ with $\mathbb{P}\{k_j = r\} = \mu(S_r)$ for all j, r and applying Hoeffding's inequality on each coordinate of the matrix $\frac{1}{d} \sum_{s=1}^d \mathcal{M}(y(k_s)) - \mathcal{M}(x)$. Then, the conclusion follows by applying a union bound. Moreover (in relation to Corollary 13), this argument can be derandomized by a classical potential-following algorithm, see [41, Lecture 4].

The additive version of the Johnson–Lindenstrauss lemma, first observed in [29] as a consequence of a deep matrix deviation inequality (see also [42, Chapter 9]), asserts that for every n points x_1, \dots, x_n in a Hilbert space \mathcal{H} and every $\varepsilon \in (0, 1)$, there exists $d \leq \frac{C \log n}{\varepsilon^2}$ and points $y_1, \dots, y_n \in \ell_2^d$ such that

$$\forall i, j \in \{1, \dots, n\}, \quad \|x_i - x_j\|_{\mathcal{H}} - \varepsilon \leq \|y_i - y_j\|_{\ell_2^d} \leq \|x_i - x_j\|_{\mathcal{H}} + \varepsilon, \quad (33)$$

where $C \in (0, \infty)$ is a universal constant. We will now observe that the spherical symmetry of \mathbf{B}_{ℓ_2} allows us to deduce a similar conclusion for points in $\mathbf{B}_{\mathcal{H}}$ by removing the incompressibility assumption from Proposition 7 when $p = 2$. We shall use the standard notation L_p^N for the space $L_p(\mu_N)$ where μ_N is the normalized counting measure on the finite set $\{1, \dots, N\}$, that is

$$\forall a = (a_1, \dots, a_N) \in \mathbb{R}^N, \quad \|a\|_{L_p^N} \stackrel{\text{def}}{=} \left(\frac{1}{N} \sum_{i=1}^N |a_i|^p \right)^{1/p}. \quad (34)$$

Observe that for $0 < p < q \leq \infty$, we have $\mathbf{B}_{L_q^N} \subseteq \mathbf{B}_{L_p^N}$.

► **Corollary 9.** *There exists a universal constant $C \in (0, \infty)$ such that the following statement holds. Fix $n \in \mathbb{N}$ and let $\{x_i\}_{i=1}^n$ be a family of vectors in $\mathbf{B}_{\mathcal{H}}$ for some Hilbert space \mathcal{H} . Then for every $\varepsilon \in (0, 1)$, there exists $d \in \mathbb{N}$ with $d \leq \frac{C(\log n)^3}{\varepsilon^4}$ and points $y_1, \dots, y_n \in \ell_2^d$ such that*

$$\forall i, j \in \{1, \dots, n\}, \quad \|x_i - x_j\|_{\mathcal{H}} - \varepsilon \leq \|y_i - y_j\|_{\ell_2^d} \leq \|x_i - x_j\|_{\mathcal{H}} + \varepsilon. \tag{35}$$

Before proceeding to the derivation of (35) we emphasize that since the given points $\{x_i\}_{i=1}^n$ belong in $\mathbf{B}_{\mathcal{H}}$, Corollary 9 is formally weaker than the Johnson–Lindenstrauss lemma. However we include it here since it differs from [20] in that the low-dimensional point set $\{y_i\}_{i=1}^n$ is not obtained as an image of $\{x_i\}_{i=1}^n$ under a typical low-rank matrix from a specific ensemble.

Proof of Corollary 9. Since any n -point subset $\{x_1, \dots, x_n\}$ of \mathcal{H} embeds linearly and isometrically in L_2^n , we assume that $x_1, \dots, x_n \in \mathbf{B}_{L_2^n}$. We will need the following claim.

▷ **Claim.** Suppose that X_1, \dots, X_n are (not necessarily independent) random vectors, each uniformly distributed on the unit sphere \mathbb{S}^{n-1} of L_2^n . Then, for some universal constant $S \in (0, \infty)$,

$$\mathbb{E}\left[\max_{i \in \{1, \dots, n\}} \|X_i\|_{L_\infty^n}\right] \leq S\sqrt{\log n}, \tag{36}$$

Proof. By a standard estimate of Schechtman and Zinn [40, Theorem 3], for a uniformly distributed random vector X on the unit sphere \mathbb{S}^{n-1} of L_2^n , we have

$$\forall t \geq \gamma_1\sqrt{\log n}, \quad \mathbb{P}\{\|X\|_{L_\infty^n} > t\} \leq e^{-\gamma_2 t^2} \tag{37}$$

for some absolute constants $\gamma_1, \gamma_2 \in (0, \infty)$. Let $W \stackrel{\text{def}}{=} \max_{i \in \{1, \dots, n\}} \|X_i\|_{L_\infty^n}$ and notice that

$$\forall K \in (\gamma_1, \infty), \quad \mathbb{E}[W] = \int_0^\infty \mathbb{P}\{W > t\} dt \leq K\sqrt{\log n} + \int_{K\sqrt{\log n}}^\infty \mathbb{P}\{W > t\} dt. \tag{38}$$

By the union bound, we have

$$\forall t > 0, \quad \mathbb{P}\{W > t\} \leq \sum_{i=1}^n \mathbb{P}\{X_i > t\} = n\mathbb{P}\{X_1 > t\}. \tag{39}$$

Combining (38) and (39), we therefore get

$$\begin{aligned} \mathbb{E}[W] &\leq K\sqrt{\log n} + n \int_{K\sqrt{\log n}}^\infty \mathbb{P}\{X_1 > t\} dt \stackrel{(37)}{\leq} K\sqrt{\log n} + n \int_{K\sqrt{\log n}}^\infty e^{-\gamma_2 t^2} dt \\ &= K\sqrt{\log n} + n\sqrt{\log n} \int_K^\infty n^{-\gamma_2 u^2} du = K\sqrt{\log n} + \sqrt{\log n} \int_K^\infty n^{1-\gamma_2 u^2} du. \end{aligned} \tag{40}$$

Choosing $K > \gamma_1$ such that $K^2\gamma_2 > 1$, the exponent in the last integrand becomes negative, thus

$$\mathbb{E}[W] \leq K\sqrt{\log n} + 2\sqrt{\log n} \int_K^\infty 2^{-\gamma_2 u^2} du \leq S\sqrt{\log n} \tag{41}$$

for a large enough constant $S \in (0, \infty)$ and the claim follows. ◁

40:10 Dimension Reduction for Incompressible Subsets of ℓ_p

Now let $U \in \mathcal{O}(n)$ be a uniformly chosen random rotation on \mathbb{R}^n . The aforementioned claim shows that since $\|x_i\|_{L_2^n} \leq 1$ for every $i \in \{1, \dots, n\}$, writing $\hat{x}_i = \frac{x_i}{\|x_i\|_{L_2^n}}$, we have the estimate

$$\mathbb{E}\left[\max_{i \in \{1, \dots, n\}} \|Ux_i\|_{L_\infty^n}\right] \leq \mathbb{E}\left[\max_{i \in \{1, \dots, n\}} \|U\hat{x}_i\|_{L_\infty^n}\right] \leq S\sqrt{\log n}. \quad (42)$$

Therefore, by (42) and Proposition 7 there exists a constant $C \in (0, \infty)$ and a rotation $U \in \mathcal{O}(n)$ such that for every $\varepsilon \in (0, 1)$ there exists $d \leq \frac{C(\log n)^3}{\varepsilon^4}$ and points $y_1, \dots, y_n \in \ell_2^d$ for which

$$\forall i, j \in \{1, \dots, n\}, \quad \|Ux_i - Ux_j\|_{L_2^n}^2 - \varepsilon^2 \leq \|y_i - y_j\|_{\ell_2^d}^2 \leq \|Ux_i - Ux_j\|_{L_2^n}^2 + \varepsilon^2. \quad (43)$$

Since $\|Ua - Ub\|_{L_2^n} = \|a - b\|_{L_2^n}$ for every $a, b \in L_2^n$, the conclusion follows by the elementary inequality $|\alpha - \beta| \leq \sqrt{|\alpha^2 - \beta^2|}$ which holds for every positive numbers $\alpha, \beta \in (0, \infty)$. ◀

► **Remark 10.** Fix $p \in [1, \infty)$. The isometric embedding theorem of Ball [6] asserts that any n -point subset of ℓ_p admits an isometric embedding into ℓ_p^N where $N = \binom{n}{2} + 1$. Suppose, more generally, that $n, N \in \mathbb{N}$ are such that N is polynomial in n . Considerations in the spirit of the proof of Corollary 9 (e.g. relying on [40]) then show that if x_1, \dots, x_n are independent uniformly random points in $\mathbf{B}_{L_p^N}$, then the random set $\{x_1, \dots, x_n\}$ is $O(\log n)^{1/p}$ -incompressible. In other words, incompressibility is a *generic* property of random n -point subsets of $\mathbf{B}_{L_p^N}$. On the other hand, a typical n -point subset of $\mathbf{B}_{L_p^N}$ is known to be approximately a simplex due to work of Arias-de-Reyna, Ball and Villa [4] and so, in particular, it can be bi-Lipschitzly embedded in $O(\log n)$ dimensions.

2.3 Factorization and proof of Theorem 2

Observe that Proposition 7 is rather non-canonical as the conclusion depends on the pairwise distances between the points $\{x_i\}_{i=1}^n$ in $L_p(\mu)$ whereas the bound on the dimension depends on $L = \max_i \|x_i\|_{L_\infty(\mu)}$. In order to deduce Theorem 2 from this (a priori weaker) statement we shall leverage the fact that Proposition 7 holds for *any* probability measure μ by optimizing this parameter L over all lattice-isomorphic images of $\{x_i\}_{i=1}^n$. The optimal such *change of measure* which allows us to replace L by $\|\max_i |x_i|\|_{L_p(\mu)}$ is a special case of a classical factorization theorem of Maurey (see [32] or [22, Theorem 5] for the general statement), whose short proof we include for completeness.

► **Proposition 11.** Fix $n \in \mathbb{N}$, $p \in (0, \infty)$ and a probability space (Ω, μ) . For every points $x_1, \dots, x_n \in L_p(\mu)$, there exists a nonnegative density function $f : \Omega \rightarrow \mathbb{R}_+$ supported on the support of $\max_i |x_i|$ such that if ν is the probability measure on Ω given by $\frac{d\nu}{d\mu} = f$, then

$$\max_{i \in \{1, \dots, n\}} \|x_i f^{-1/p}\|_{L_\infty(\nu)} \leq \left\| \max_{i \in \{1, \dots, n\}} |x_i| \right\|_{L_p(\mu)}. \quad (44)$$

Proof. Let $V = \text{supp}(\max_i |x_i|) \subseteq \Omega$ and define the change of measure f as

$$\forall \omega \in \Omega, \quad f(\omega) \stackrel{\text{def}}{=} \frac{\max_{i \in \{1, \dots, n\}} |x_i(\omega)|^p}{\int_\Omega \max_{i \in \{1, \dots, n\}} |x_i(\theta)|^p d\theta}. \quad (45)$$

Then, (44) is elementary to check. ◀

We are now ready to complete the proof of Theorem 2.

Proof of Theorem 2. Fix a K -incompressible family of vectors $x_1, \dots, x_n \in L_p(\Omega, \mu)$ and let $V = \text{supp}(\max_i |x_i|) \subseteq \Omega$. Denote by $f : \Omega \rightarrow \mathbb{R}_+$ the change of density from Proposition 11. If $\frac{d\nu}{d\mu} = f$, then the linear operator $T : L_p(V, \mu) \rightarrow L_p(\Omega, \nu)$ given by $Tg = f^{-1/p}g$ is (trivially) a linear isometry. Therefore, Proposition 7 and (44) show that there exists $d \in \mathbb{N}$ with $d \leq \frac{32e^2(2K)^{2p} \log n}{\varepsilon^2}$ and points $y_1, \dots, y_n \in \ell_p^d$ such that the condition

$$\begin{aligned} \|x_i - x_j\|_{L_p(\mu)}^p - \varepsilon &= \|Tx_i - Tx_j\|_{L_p(\nu)}^p - \varepsilon \\ &\leq \|y_i - y_j\|_{\ell_p^d}^p \leq \|Tx_i - Tx_j\|_{L_p(\nu)}^p + \varepsilon = \|x_i - x_j\|_{L_p(\mu)}^p + \varepsilon, \end{aligned} \tag{46}$$

is satisfied for every $i, j \in \{1, \dots, n\}$. This concludes the proof of Theorem 2. \blacktriangleleft

► Remark 12. A careful inspection of the proof of Theorem 2 reveals that the low-dimensional points $\{y_i\}_{i=1}^n$ can be obtained as images of the given points $\{x_i\}_{i=1}^n$ under a linear transformation. Indeed, starting from a K -incompressible family of points $\{x_i\}_{i=1}^n$ in $L_p(\Omega, \mu)$, we use Proposition 11 to find a change of measure $T : L_p(V, \mu) \rightarrow L_p(\Omega, \nu)$ such that $\{Tx_i\}_{i=1}^n$ satisfy the stronger assumption of Proposition 7. Then, for some $d \in \mathbb{N}$ with $d \leq \frac{32e^2(2K)^{2p} \log n}{\varepsilon^2}$ we find pairwise disjoint measurable subsets S_1, \dots, S_d of Ω , each with positive measure, such that if $S : L_p(\Omega, \nu) \rightarrow \ell_p^d$ is the linear map

$$\forall z \in L_p(\Omega, \nu), \quad Sz \stackrel{\text{def}}{=} \frac{1}{d^{1/p}} \left(\frac{1}{\mu(S_1)} \int_{S_1} z \, d\nu, \dots, \frac{1}{\mu(S_d)} \int_{S_d} z \, d\nu \right) \in \ell_p^d, \tag{47}$$

then the points $\{y_i\}_{i=1}^n = \{(S \circ T)x_i\}_{i=1}^n \subseteq \ell_p^d$ satisfy the desired conclusion (7).

We conclude this section by observing that the argument leading to Theorem 2 is constructive.

► Corollary 13. *In the setting of Theorem 2, there exists a greedy algorithm which receives as input the high-dimensional points $\{x_i\}_{i=1}^n$ and produces as output the low-dimensional points $\{y_i\}_{i=1}^n$.*

Proof. As the density (45) is explicitly defined, the linear operator $T : L_p(V, \mu) \rightarrow L_p(\Omega, \nu)$ can also be efficiently constructed. On the other hand, in order to construct the operator S defined by (47) one needs to find the corresponding partition $\{S_1, \dots, S_d\}$ and this was achieved in Proposition 7 via an application of Maurey’s sampling lemma to the cone $\mathcal{C}_p \subseteq \ell_\infty^N$ where $N = \binom{n}{2}$. As ℓ_∞^N is ε -isomorphic to the 2-uniformly smooth space $\ell_{\log N}^N$, Ivanov’s result from [17] implies that the construction can be implemented by a greedy algorithm. \blacktriangleleft

3 Proof of Theorem 6

In this section we prove Theorem 6. The constructed subset of \mathbf{B}_{ℓ_p} which does not embed linearly into ℓ_p^d for small d is a slight modification of the one considered in [27].

Proof of Theorem 6. Fix $m \in \mathbb{N}$ and denote by $\{w_i\}_{i=1}^{2^m}$ the rows of the $2^m \times 2^m$ Walsh matrix and by $\{e_i\}_{i=1}^{2^m}$ the coordinate basis vectors of \mathbb{R}^{2^m} . Consider the n -point set

$$\mathcal{S}_{n,p} = \{0\} \cup \{e_1, \dots, e_{2^m}\} \cup \left\{ \frac{w_1}{2^{m/p}}, \dots, \frac{w_{2^m}}{2^{m/p}} \right\} \subseteq \mathbf{B}_{\ell_p^{2^m}} \tag{48}$$

where $n = 2^{m+1} + 1$ and suppose that $T : \ell_p^{2^m} \rightarrow \ell_p^d$ is a linear operator such that

$$\forall x, y \in \mathcal{S}_{n,p}, \quad \omega(\|x - y\|_{\ell_p^{2^m}}) \leq \|Tx - Ty\|_{\ell_p^d} \leq \Omega(\|x - y\|_{\ell_p^{2^m}}). \tag{49}$$

40:12 Dimension Reduction for Incompressible Subsets of ℓ_p

Assume first that $1 \leq p < 2$. If we write $w_i = \sum_{j=1}^{2^m} w_i(j)e_j$ then by orthogonality of $\{w_i\}_{i=1}^{2^m}$,

$$\sum_{i=1}^{2^m} \|Tw_i\|_{\ell_2^d}^2 = \sum_{i=1}^{2^m} \left\| \sum_{j=1}^{2^m} w_i(j)Te_j \right\|_{\ell_2^d}^2 = \sum_{j,k=1}^{2^m} \langle w_j, w_k \rangle \langle Te_j, Te_k \rangle = 2^m \sum_{j=1}^{2^m} \|Te_j\|_{\ell_2^d}^2. \quad (50)$$

By assumption (49) on T , we have

$$\forall j \in \{1, \dots, 2^m\}, \quad \|Te_j\|_{\ell_2^d}^2 \leq \|Te_j\|_{\ell_p^d}^2 \leq \Omega(1)^2 \quad (51)$$

and

$$\forall j \in \{1, \dots, 2^m\}, \quad \|Tw_j\|_{\ell_2^d}^2 \geq 2^{\frac{2m}{p}} d^{-\frac{2-p}{p}} \left\| T\left(\frac{w_j}{2^{m/p}}\right) \right\|_{\ell_p^d}^2 \geq 2^{\frac{2m}{p}} d^{-\frac{2-p}{p}} \omega(1)^2. \quad (52)$$

Combining (50), (51) and (52) we deduce that

$$2^{m(1+\frac{2}{p})} d^{-\frac{2-p}{p}} \omega(1)^2 \leq 4^m \Omega(1)^2, \quad (53)$$

which is equivalent to $d \geq \left(\frac{\omega(1)}{\Omega(1)}\right)^{\frac{2p}{2-p}} 2^m = \left(\frac{\omega(1)}{\Omega(1)}\right)^{\frac{2p}{|p-2|}} \cdot \frac{n-1}{2}$. The case $p > 2$ is treated similarly. \blacktriangleleft

References

- 1 D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *J. Comput. System Sci.*, 66(4):671–687, 2003. Special issue on PODS 2001 (Santa Barbara, CA). doi:10.1016/S0022-0000(03)00025-4.
- 2 N. Ailon and B. Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009. doi:10.1137/060673096.
- 3 N. Ailon and E. Liberty. An almost optimal unrestricted fast Johnson-Lindenstrauss transform. *ACM Trans. Algorithms*, 9(3):Art. 21, 12, 2013. doi:10.1145/2483699.2483701.
- 4 J. Arias-de Reyna, K. Ball, and R. Villa. Concentration of the distance in finite-dimensional normed spaces. *Mathematika*, 45(2):245–252, 1998. doi:10.1112/S0025579300014182.
- 5 R. I. Arriaga and S. Vempala. An algorithmic theory of learning: robust concepts and random projection. In *40th Annual Symposium on Foundations of Computer Science (New York, 1999)*, pages 616–623. IEEE Computer Soc., Los Alamitos, CA, 1999. doi:10.1109/SFFCS.1999.814637.
- 6 K. Ball. Isometric embedding in ℓ_p -spaces. *European J. Combin.*, 11(4):305–311, 1990. doi:10.1016/S0195-6698(13)80131-X.
- 7 S. Barman. Approximating Nash equilibria and dense subgraphs via an approximate version of Carathéodory’s theorem. *SIAM J. Comput.*, 47(3):960–981, 2018. doi:10.1137/15M1050574.
- 8 Y. Bartal and L.-A. Gottlieb. Approximate nearest neighbor search for ℓ_p -spaces ($2 < p < \infty$) via embeddings. In *LATIN 2018: Theoretical informatics*, volume 10807 of *Lecture Notes in Comput. Sci.*, pages 120–133. Springer, Cham, 2018. doi:10.1007/978-3-319-77404-6_1.
- 9 J. Bourgain, S. Dirksen, and J. Nelson. Toward a unified theory of sparse dimensionality reduction in Euclidean space. *Geom. Funct. Anal.*, 25(4):1009–1088, 2015. doi:10.1007/s00039-015-0332-9.
- 10 B. Brinkman and M. Charikar. On the impossibility of dimension reduction in l_1 . *J. ACM*, 52(5):766–788, 2005. doi:10.1145/1089023.1089026.
- 11 S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures Algorithms*, 22(1):60–65, 2003. doi:10.1002/rsa.10073.
- 12 S. Dirksen. Dimensionality reduction with subgaussian matrices: a unified theory. *Found. Comput. Math.*, 16(5):1367–1396, 2016. doi:10.1007/s10208-015-9280-x.

- 13 P. Frankl and H. Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *J. Combin. Theory Ser. B*, 44(3):355–362, 1988. doi:10.1016/0095-8956(88)90043-3.
- 14 Y. Gordon. On Milman’s inequality and random subspaces which escape through a mesh in \mathbf{R}^n . In *Geometric aspects of functional analysis (1986/87)*, volume 1317 of *Lecture Notes in Math.*, pages 84–106. Springer, Berlin, 1988. doi:10.1007/BFb0081737.
- 15 P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC ’98 (Dallas, TX)*, pages 604–613. ACM, New York, 1999.
- 16 P. Indyk and A. Naor. Nearest-neighbor-preserving embeddings. *ACM Trans. Algorithms*, 3(3):Art. 31, 12, 2007. doi:10.1145/1273340.1273347.
- 17 G. Ivanov. Approximate Carathéodory’s Theorem in Uniformly Smooth Banach Spaces. *Discrete Comput. Geom.*, 66(1):273–280, 2021. doi:10.1007/s00454-019-00130-w.
- 18 L. Jacques. A quantized Johnson-Lindenstrauss lemma: the finding of Buffon’s needle. *IEEE Trans. Inform. Theory*, 61(9):5012–5027, 2015. doi:10.1109/TIT.2015.2453355.
- 19 L. Jacques. Small width, low distortions: quantized random embeddings of low-complexity sets. *IEEE Trans. Inform. Theory*, 63(9):5477–5495, 2017.
- 20 W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemp. Math.*, pages 189–206. Amer. Math. Soc., Providence, RI, 1984. doi:10.1090/conm/026/737400.
- 21 W. B. Johnson and A. Naor. The Johnson-Lindenstrauss lemma almost characterizes Hilbert space, but not quite. *Discrete Comput. Geom.*, 43(3):542–553, 2010. doi:10.1007/s00454-009-9193-z.
- 22 W. B. Johnson and G. Schechtman. Finite dimensional subspaces of L_p . In *Handbook of the geometry of Banach spaces, Vol. I*, pages 837–870. North-Holland, Amsterdam, 2001. doi:10.1016/S1874-5849(01)80021-8.
- 23 D. M. Kane and J. Nelson. Sparser Johnson-Lindenstrauss transforms. *J. ACM*, 61(1):Art. 4, 23, 2014. doi:10.1145/2559902.
- 24 B. Klartag and S. Mendelson. Empirical processes and random projections. *J. Funct. Anal.*, 225(1):229–245, 2005. doi:10.1016/j.jfa.2004.10.009.
- 25 F. Krahermer and R. Ward. New and improved Johnson-Lindenstrauss embeddings via the restricted isometry property. *SIAM J. Math. Anal.*, 43(3):1269–1281, 2011. doi:10.1137/100810447.
- 26 M. Ledoux and M. Talagrand. *Probability in Banach spaces*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*. Springer-Verlag, Berlin, 1991. Isoperimetry and processes. doi:10.1007/978-3-642-20212-4.
- 27 J. R. Lee, M. Mendel, and A. Naor. Metric structures in L_1 : dimension, snowflakes, and average distortion. *European J. Combin.*, 26(8):1180–1190, 2005. doi:10.1016/j.ejc.2004.07.002.
- 28 J. R. Lee and A. Naor. Embedding the diamond graph in L_p and dimension reduction in L_1 . *Geom. Funct. Anal.*, 14(4):745–747, 2004. doi:10.1007/s00039-004-0473-8.
- 29 C. Liaw, A. Mehrabian, Y. Plan, and R. Vershynin. A simple tool for bounding the deviation of random matrices on geometric sets. In *Geometric aspects of functional analysis*, volume 2169 of *Lecture Notes in Math.*, pages 277–299. Springer, Cham, 2017.
- 30 J. Matoušek. On the distortion required for embedding finite metric spaces into normed spaces. *Israel J. Math.*, 93:333–344, 1996. doi:10.1007/BF02761110.
- 31 J. Matoušek. On variants of the Johnson-Lindenstrauss lemma. *Random Structures Algorithms*, 33(2):142–156, 2008. doi:10.1002/rsa.20218.
- 32 B. Maurey. *Théorèmes de factorisation pour les opérateurs linéaires à valeurs dans les espaces L^p* . Astérisque, No. 11. Société Mathématique de France, Paris, 1974. With an English summary.
- 33 A. Naor. Metric dimension reduction: a snapshot of the Ribe program. In *Proceedings of the International Congress of Mathematicians—Rio de Janeiro 2018. Vol. I. Plenary lectures*, pages 759–837. World Sci. Publ., Hackensack, NJ, 2018.

- 34 A. Naor, G. Pisier, and G. Schechtman. Impossibility of dimension reduction in the nuclear norm. *Discrete Comput. Geom.*, 63(2):319–345, 2020. doi:10.1007/s00454-019-00162-2.
- 35 M. I. Ostrovskii. *Metric embeddings*, volume 49 of *De Gruyter Studies in Mathematics*. De Gruyter, Berlin, 2013. Bilipschitz and coarse embeddings into Banach spaces. doi:10.1515/9783110264012.
- 36 G. Pisier. Remarques sur un résultat non publié de B. Maurey. In *Seminar on Functional Analysis, 1980–1981*, pages Exp. No. V, 13. École Polytech., Palaiseau, 1981.
- 37 Y. Plan and R. Vershynin. Dimension reduction by random hyperplane tessellations. *Discrete Comput. Geom.*, 51(2):438–461, 2014. doi:10.1007/s00454-013-9561-6.
- 38 O. Regev and T. Vidick. Bounds on dimension reduction in the nuclear norm. In *Geometric aspects of functional analysis. Vol. II*, volume 2266 of *Lecture Notes in Math.*, pages 279–299. Springer, Cham, [2020] ©2020. doi:10.1007/978-3-030-46762-3_13.
- 39 G. Schechtman. Two observations regarding embedding subsets of Euclidean spaces in normed spaces. *Adv. Math.*, 200(1):125–135, 2006. doi:10.1016/j.aim.2004.11.003.
- 40 G. Schechtman and J. Zinn. On the volume of the intersection of two L_p^n balls. *Proc. Amer. Math. Soc.*, 110(1):217–224, 1990. doi:10.2307/2048262.
- 41 J. Spencer. *Ten lectures on the probabilistic method*, volume 64 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 1994. doi:10.1137/1.9781611970074.
- 42 R. Vershynin. *High-dimensional probability*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2018. An introduction with applications in data science, With a foreword by Sara van de Geer. doi:10.1017/9781108231596.

Short Topological Decompositions of Non-Orientable Surfaces

Niloufar Fuladi ✉

LIGM, CNRS, Univ. Gustave Eiffel, ESIEE Paris, F-77454 Marne-la-Vallée, France

Alfredo Hubard

LIGM, CNRS, Univ. Gustave Eiffel, ESIEE Paris, F-77454 Marne-la-Vallée, France

Arnaud de Mesmay ✉

LIGM, CNRS, Univ. Gustave Eiffel, ESIEE Paris, F-77454 Marne-la-Vallée, France

Abstract

We investigate short topological decompositions of non-orientable surfaces and provide algorithms to compute them. Our main result is a polynomial-time algorithm that for any graph embedded in a non-orientable surface computes a canonical non-orientable system of loops so that any loop from the canonical system intersects any edge of the graph in at most 30 points. The existence of such short canonical systems of loops was well known in the orientable case and an open problem in the non-orientable case. Our proof techniques combine recent work of Schaefer-Štefankovič with ideas coming from computational biology, specifically from the signed reversal distance algorithm of Hannenhalli-Pevzner. This result confirms a special case of a conjecture of Negami on the joint crossing number of two embeddable graphs. We also provide a correction for an argument of Negami bounding the joint crossing number of two non-orientable graph embeddings.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Mathematics of computing → Graphs and surfaces

Keywords and phrases Computational topology, embedded graph, non-orientable surface, joint crossing number, canonical system of loop, surface decomposition

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.41

Related Version *Full Version:* <https://arxiv.org/abs/2203.06659> [9]

Funding This work was partially supported by the ANR project SoS (ANR-17-CE40-0033).

Acknowledgements We are grateful to Marcus Schaefer and Daniel Štefankovič for providing us the full version of [25], to Francis Lazarus for insightful discussions, and to the anonymous reviewers for very helpful comments.

1 Introduction

Decomposing a surface along a graph or a curve is a standard way to simplify its topology. The classification of surfaces and classical tools to compute both homology groups and fundamental groups typically rely on such topological decompositions, which are also important in meshing and 3D-modeling (see for example [27]). Surfaces often come with extra structure which can be modeled by an embedded graph. Decomposing such a surface efficiently then means finding a graph that intersects the original graph transversely and that does not intersect the embedded graph too much but nevertheless carries the topological complexity of the surface, as is done for example in [18] or [8]. Such decompositions also appear in algorithm design: often, to generalize results on planar graph to graphs embedded on surfaces, it is enough to find a decomposition that cuts open the surface into a disk, then solve the resulting planar instance and stitch back the solution, see, e.g., [4, 7, 18]. Sometimes it is important that the cut graph is canonical in some way, e.g. in order to compute a homeomorphism between two surfaces, one cuts them into disks, puts these disks in correspondence, and glues back the surfaces. This works only if the cut graphs have the same combinatorial structure.



© Niloufar Fuladi, Alfredo Hubard, and Arnaud de Mesmay;
licensed under Creative Commons License CC-BY 4.0

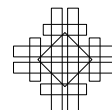
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 41; pp. 41:1–41:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Lazarus, Pocchiola, Vegter and Verroust [18] (see also [17]) were the first to design an algorithm that finds, for any graph G embedded in a closed orientable surface S a *canonical system of loops* H such that no edge of H intersects¹ any edge of G more than a constant number of times. Here, by a canonical system of loops we mean a one-vertex and one-face embedded graph in which the cyclic ordering of the edges around the vertex is $a_1 b_1 a_1^{-1} b_1^{-1} \dots a_g b_g a_g^{-1} b_g^{-1}$. Such a decomposition is an instance of the joint crossing number problem. More precisely, consider a pair of graphs G_1 and G_2 embedded on a surface S of genus g and define the *joint crossing number* as the minimal number of crossings between $h(G_1)$ and G_2 over all the homeomorphisms $h : S \rightarrow S$. This quantity was introduced by Negami [22] who made the following conjecture:

► **Conjecture 1.** *There exists a universal constant C such that for any pair of graphs G_1 and G_2 embedded on a surface S , the joint crossing number is at most $C|E(G_1)||E(G_2)|$.*

Furthermore, he proved an upper bound of $Cg|E(G_1)||E(G_2)|$. His conjecture has been investigated further [1, 15, 23] and variants of this problem have appeared in works with applications as diverse as finding explicit bounds for graph minors [10] or designing an algorithm for the embeddability of simplicial complexes into \mathbb{R}^3 [19].

In the non-orientable case, no instance of Negami’s conjecture seem to be known. Even if G_1 is the *non-orientable canonical system of loops*, that is, a system of one-sided loops with the cyclic ordering $a_1 a_1 a_2 a_2 \dots a_g a_g$ around the vertex, the best known bound is $O(g|E(G_2)|)$ crossings for each loop (see [17]), which matches Negami’s bound. Non-orientable surfaces have been often somehow neglected in computational topology, but there are many reasons to want to correct this: a random surface that arises from pasting a set of polygons along their edges is non-orientable with overwhelming probability, they appear as configuration spaces in diverse contexts [11, 26], and insights garnered from non-orientable surfaces can sometimes also be applied to the orientable ones; see for example [24]. Furthermore, the orientable genus of a graph can be arbitrarily larger than its non-orientable genus, while the reverse does not happen (see Lemma 7).

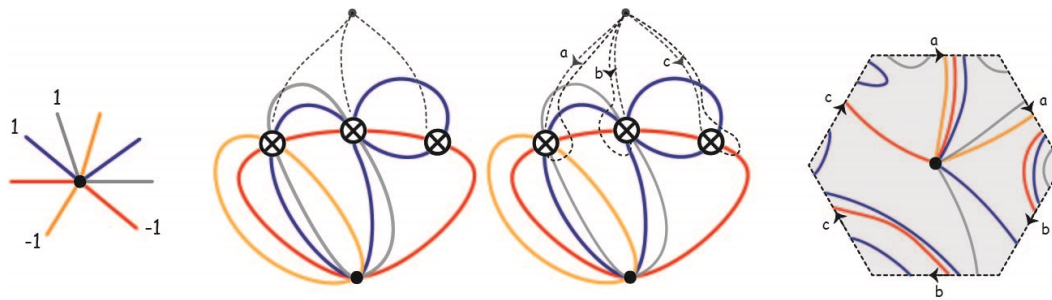
Our results. In this article, we initiate a study of short topological decompositions on non-orientable surfaces. We first show that the proof of the aforementioned result of Negami [22] has a minor flaw in the non-orientable case, we provide a counter-example to the proof technique and an alternative proof based on different techniques.

► **Theorem 2.** *Let S be a non-orientable surface of genus $g \geq 1$ and G_1 and G_2 be two graphs embedded on S . Then there exists a homeomorphism h such that any edge of $h(G_1)$ crosses each edge of G_2 at most $O(g)$ times. In particular, the total number of crossings between $h(G_1)$ and G_2 is $O(g|E(G_1)||E(G_2)|)$.*

Then our main result is the following theorem providing, to the best of our knowledge, the first known case of a short topological decomposition into a disk for non-orientable surfaces.

► **Theorem 3.** *There exists a polynomial time algorithm that given a graph cellularly embedded on a non-orientable surface computes a non-orientable canonical system of loops such that each loop in the system intersects any edge of the graph in at most 30 points.*

¹ Throughout the article, we decompose surface-embedded graphs by cutting them along embedded graphs which are transverse to the original graph, and count the number of intersections. This is equivalent to the primal setting studied in, e.g., Lazarus, Pocchiola, Vegter and Verroust [18] via graph duality.



■ **Figure 1** From left to right: 1) The combinatorial information of a one-vertex graph. 2) A cross-cap drawing of this graph, with cross-caps connected to a base-point. 3) A joint drawing of the graph and a canonical system of loops. 4) A different representation: decomposing the graph along the canonical system of loops.

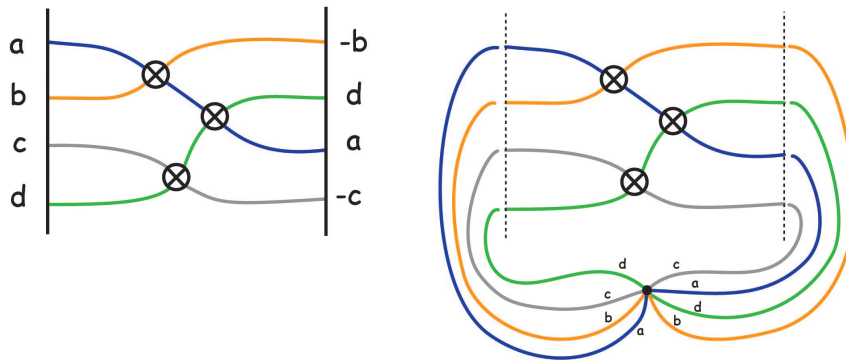
Main ideas and proof techniques. As in many similar works, the first step in most of our results is to contract a spanning tree of the underlying graph, reducing the problems to the setting of one-vertex graph embedded on a non-orientable surface. The combinatorics of a one-vertex embedded graph are completely described by a rotation system, or embedding scheme. This will be the basic object with which we work.

Then, a simple but important object that we rely on extensively is an *orienting curve*, i.e., a closed curve on a non-orientable surface such that cutting along it produces an orientable surface. It was shown by Matoušek, Sedgwick, Tancer and Wagner [19] that given a graph embedded on a non-orientable surface, one can compute such an orienting loop that crosses each edge of the graph at most twice. This tool will be crucial in our corrected proof of Theorem 2 and in our work to prove Theorem 3.

For the proof of Theorem 3, we first point out that the techniques used to prove the orientable version in [18] incur an overhead of $O(g)$ in the number of crossings of the resulting curves (see [17, Theorem 4.3.9]). Instead, our proof of Theorem 3 builds on important recent work of Schaefer and Štefankovič [25]. The foundational idea behind this work, which takes its roots in an article of Mohar [20] on the degenerate crossing number, is to represent a graph embedded on a non-orientable surface as a planar drawing, with a finite set of *cross-caps*, as pictured in the second image of Figure 1. Schaefer and Štefankovič showed that any graph embedded on a non-orientable surface can be represented with a cross-cap drawing so that each edge uses each cross-cap at most twice. Our main technical contribution is to upgrade their construction so that the cross-caps can be connected to each other so as to yield a non-orientable canonical system of loops (Lemma 8), so that each loop intersects each edge of the one-vertex graph in at most 30 points (see Figure 1).

The complexity of the drawings provided by the proof of Schaefer and Štefankovič increases too fast to directly obtain a good bound by just connecting the cross-caps. Therefore, we modify their algorithm. First, by the aforementioned techniques, we can assume that we always have an orienting loop, which simplifies some of the steps and provides additional structure to the inductive argument. More importantly, we show that one can impose a certain order in which we choose the one-sided loops, as well as the separating loops, in the inductive argument to obtain a finer control on the resulting drawing.

The order in which we choose the loops comes from a seemingly unrelated problem in computational biology, and more precisely genome rearrangements. Given a permutation with signatures (a bit assigned to each letter), a signed reversal consists in choosing a subword in w , and reversing it as well as the signatures of all its letters. The *signed reversal distance*



■ **Figure 2** Left: a pictorial representation of three signed reversals bringing the signed permutation on the left to the signed permutation of the right. Right: Attaching the two permutations to a common basepoint yields a one-vertex graph with an embedding scheme, and the signed reversals provide a cross-cap drawing of that scheme where each loop enters each cross-cap at most once.

between two signed permutations is the minimum number of signed reversals needed to go from one permutation to the other one. This distance, and in particular algorithms to compute it has been intensively studied in the computational biology literature due to its relevance for phylogenetic reconstruction (see for example [14]). A cornerstone of the theory is the breakthrough of Hannenhalli and Pevzner [12] who provided an algorithm to compute the signed reversal distance between two signed permutations in polynomial time (see also the reformulation by Bergeron [2]). Now, as we illustrate in Figure 2, there is a very strong similarity between computing the signed reversal distance between two permutations and embedding a one-vertex graph built from these two permutations with a minimum number of cross-caps (see also [3, 16]). Surprisingly the algorithms of Hannenhalli and Pevzner on one side and of Schaefer and Štefankovič on the other also have strong similarities, which we leverage for example in Lemma 22. We hope that further cross-pollination between computational genomics and computational topology will lead to new surprises.

Due to space limitations, many proofs are omitted but can be found in the full version [9].

2 Preliminaries

We refer the reader to standard references such as Hatcher [13] and Stillwell [28] for topological background, the book of Mohar and Thomassen for graphs on surfaces [21] and the survey of Colin de Verdière [5] for topological algorithms for embedded graphs.

Surfaces, curves and embedded graphs. We assume working knowledge with the classification of surfaces. By the *genus* of a surface, we mean the orientable genus for an orientable surface, which we denote by M , and the non-orientable genus for a non-orientable surface, which we denote by N . The Euler genus is twice the orientable genus for orientable surfaces $eg(M) = 2g(M)$, and equals the non-orientable genus for non-orientable ones $eg(N) = g(N)$.

We call a closed curve on a surface *two-sided* if it has a neighborhood of it homeomorphic to the annulus. Otherwise, it is called *one-sided* and it has a neighborhood homeomorphic to the Möbius band. Given a closed curve ν on a surface S , cutting S along ν gives a (possibly disconnected) surface with one or two boundary components depending on whether ν is one-sided or two-sided. A curve δ on a surface S is *non-separating* if the surface we obtain by cutting along δ is connected; otherwise δ is separating. An *orienting curve* on a non-orientable surface N is a curve γ such that by cutting along γ , we get a connected orientable surface. We recall the following lemma from [19, Lemma 5.3].

► **Lemma 4.** *Let N be a non-orientable surface of genus g with h boundary components and let γ be an orienting closed curve. Let g_γ be the (orientable) genus and h_γ be the number of boundary components in N after cutting along γ .*

- *If g is odd, then γ is one-sided, $g_\gamma = \frac{g-1}{2}$, and $h_\gamma = h + 1$*
- *If g is even, then γ is two-sided, $g_\gamma = \frac{g-2}{2}$, and $h_\gamma = h + 2$.*

On a surface with boundary, by an *essential* proper arc, we mean an arc with endpoints on a boundary component that does not cut off a disk from the surface.

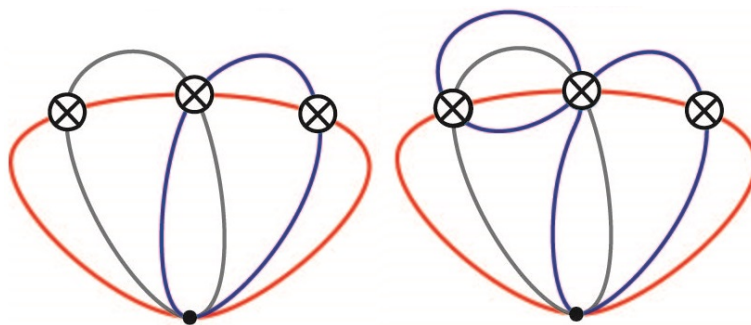
An *embedding* of a graph G on a surface S is a continuous injective map from G into S . A graph embedding is called *cellular* if its faces are homeomorphic to open disks.

Discrete Metrics on Surfaces. A cellularly embedded graph G on a surface S induces a discrete metric in two different ways. In the *combinatorial model*, the metric is defined on walks in G , and the length of a curve C is the number of edges of G traversed by C . In the *cross-metric model* [6], the metric is defined on curves *transverse* to G (i.e., that intersect G only at edges and in a non-tangent way), and their lengths is the number of crossings with G . Both models are naturally connected via graph duality. We mostly work in the cross-metric model and we refer to the embedded graph G of a cross metric surface S as the primal graph on S . The *multiplicity* of a curve (or a system of curves) at some edge e of G is the number of times e is crossed by the curve (curves). The multiplicity of a curve (or a system of curves) is the maximal multiplicity of the curve (curves) at any edge e of G .

Embedding schemes. For v a vertex of an embedded graph G , by a *rotation* ρ_v at v , we mean the cyclic permutation of the ends of edges incident to v . A *rotation system*, ρ , of a graph assigns a rotation to each vertex. and a signature to each edge, which is a number from $\{1, -1\}$. A rotation system ρ and a signature λ for the edges determine a cellular embedding for the graph up to homeomorphism i.e. we can compute the faces of the embedding purely combinatorially (see [21, Section 3.2] for further details). The pair (ρ, λ) is called an *embedding scheme* for the graph G , we simply use *scheme* throughout this work. Since a first step in all of our arguments is to contract a spanning tree, almost all the schemes considered in this article will have a single vertex. A cycle in a scheme is one-sided if the signature of its edges multiply to -1 and it is two-sided otherwise. The Euler genus of a scheme is the Euler genus of the underlying surface. A scheme is *orientable* if all its cycles are two-sided, and *non-orientable* otherwise. A loop e in the scheme divides the half-edges around the vertex into two parts; each part is called a *wedge* of e . When a loop γ has exactly one end in each wedge of e , we say that the ends of γ *alternate* with those of e ; otherwise both ends of γ is in one wedge of e and we say that the ends of e *enclose* the ends of γ .

Following Schaefer and Štefankovič [25], we use the following model with localized cross-caps to represent non-orientable embedded graphs. A *planarizing system of disjoint one-sided curves* on a non-orientable surface, abbreviated *PDIS*, is a system of g disjoint one-sided curves such that by cutting along them, we obtain a sphere with g holes (this was first introduced by Mohar [20]). Therefore, from any graph embedded on a non-orientable surface, we obtain a planar representation by cutting along such a system. The non-orientable surface is recovered by gluing a Möbius band on each boundary component, which we depict using \otimes and call a *cross-cap*. It is easily checked that a family of edges entering a cross-cap emerges on the other side with a reversed order, and that the sidedness of a loop is determined by the number of cross-caps that it crosses.

The planar drawing that we obtain by this cross-cap localization is called a *cross-cap drawing*, see Figure 3 for examples. In this model, we say that a drawing realizes an embedding scheme (G, ρ, λ) if the rotation at each vertex is as prescribed by ρ , and if



■ **Figure 3** Different localization for the same embedding scheme gives different cross-cap drawings.

whenever a closed curve in the drawing passes through an odd (resp. even) number of cross-caps, the multiplication of the signatures of the edges it follows is -1 (resp. 1). Note that such a realization might not correspond to a cellularly embedded graph, and that while a cross-cap drawing uniquely describes an embedded graph, the converse is not true, see Figure 3. Throughout this article, by a cross-cap drawing for a graph G with an embedding scheme (ρ, λ) , we mean the planar graph with cross-caps treated as extra vertices and edges being the sub-edges in G .

The following lemmas establish basic properties of orienting, separating loops and cross-cap drawings.

► **Lemma 5.** *A loop o in a cellularly embedded one-vertex graph G with a non-orientable embedding scheme is orienting if and only if its ends enclose the ends of any two-sided loop and alternate with the ends of any one-sided loop in the scheme.*

► **Lemma 6.** *In any cross-cap drawing of a scheme, a separating loop passes through each cross-cap an even number of times; and an orienting loop passes through each cross-cap an odd number of times.*

► **Lemma 7.** *Let G be an orientable scheme corresponding to a cellular embedding on a surface M with a minimum number of $g > 0$ handles. The minimum number of cross-caps needed in a cross-cap drawing realizing G is $2g + 1$ and this can always be achieved.*

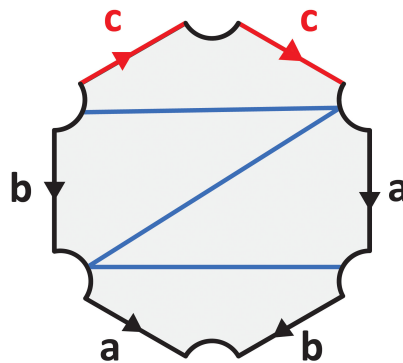
Canonical System of Loops. For a non-orientable surface of genus g , a *non-orientable canonical system of loops* is a family of one-sided loops with the cyclic ordering $a_1 a_1 a_2 a_2 \dots a_g a_g$ around the base point such that cutting M along this family yields a topological disk. The following lemma, illustrated in Figure 1 underpins our strategy to prove Theorem 3.

► **Lemma 8.** *Let H be a cross-cap drawing for a graph of non-orientable genus g and let b be a point in one face of the drawing. Let $\{p_i\}$ be a family of paths in the dual graph of this drawing from each cross-cap to b . Introduce a loop c_i by starting from b , passing along the path p_i , entering the corresponding cross-cap, going around the cross-cap and passing along p_i to return to b . The system of loops $\{c_i\}$ is a non-orientable canonical system of loops.*

Short Orienting Curves. The following lemma is a restatement of [19, Proposition 5.5].

► **Lemma 9.** *Let N be a non-orientable surface without boundary and with genus g and G be a graph embedded on N . Then there exists an orienting curve of multiplicity at most 2.*

By a little modification of the building process in the proof of this lemma, we can get an orienting arc in the case with boundaries instead of an orienting cycle, see the full version.



■ **Figure 4** A non-orientable surface of genus 3 with an embedded system of arcs. The dents in the picture indicate the segments of the boundary component.

3 Correcting the proof of Negami

Negami proved in [22, Theorem 1] that if we have two graphs embeddable on a closed surface, we can reembed them simultaneously such that their edges cross few times.

► **Theorem 10.** *Let G_1 and G_2 be two connected graphs embeddable on a closed surface of genus g , orientable or non-orientable. We can embed them simultaneously such that they intersect transversely in their edges at most $4g\beta(G_1)\beta(G_2)$ times, where $\beta(G) = |E(G)| - |V(G)| + 1$ is the Betti number for a connected graph G .*

Negami’s proof in the non-orientable case is reduced to showing the following two lemmas (with slightly different constants). However, his proof of Lemma 12 is flawed.

► **Lemma 11.** *For two orientable surfaces M_i of genus $g \geq 1$, with one boundary component and β_i disjoint essential proper arcs ($i = 1, 2$) where $\beta_i \leq \beta(G_i)$, there are homeomorphisms $\phi_i : M_i \rightarrow M$, where M is an orientable surface of genus g and one boundary component, so that the image of the arcs in M_1 and M_2 on M intersect at most $4(g - 1)\beta_1\beta_2$ times.*

► **Lemma 12.** *For two non-orientable surfaces N_i of genus $g \geq 1$ with one boundary component and β_i disjoint essential proper arcs ($i = 1, 2$) where $\beta_i \leq \beta(G_i)$, there are homeomorphisms $\phi_i : N_i \rightarrow N$, where N is a non-orientable surface of genus g and one boundary component, so that the image of the arcs in N_1 and N_2 on N intersect at most $18(g - 1)\beta_1\beta_2$ times when g is odd and $72(g - 2)\beta_1\beta_2$ when it is even.*

In the proof of Lemma 12, Negami uses induction on the genus of the non-orientable surface. Assuming that the lemma is true for genus $g - 1$, to prove it for genus g , he claims that there is an essential proper arc α that runs along the center line of a Möbius band (a one-sided arc). The idea is then to cut along α to get a non-orientable surface of genus $g - 1$ to use the induction hypothesis. The problem lies in the fact that cutting along such an arc, we might not end up with a non-orientable surface.

► **Lemma 13.** *Consider the non-orientable surface of genus 3 with one boundary component and embedded essential arcs shown in Figure 4. Any one-sided arc disjoint from the embedded arcs cut the surface into an orientable surface.*

A Correction. To prove Theorem 2, we provide a different proof of Lemma 12. Our approach is to cut both surfaces along an orienting arc guaranteed by Lemma 9, this has the effect of multiplying the number of arcs by at most three. Depending on the parity of the genus we obtain an orientable surface with one or two boundary components. In the odd case, we apply Lemma 11 and then carefully modify one of the maps near the boundary controlling the multiplicity so that we can reconstruct the surface. In the even case, we have two boundaries. In that case, we connect them using a path of controlled multiplicity. Then we argue like in the odd case.²

4 Non-orientable Canonical System of Loops

4.1 The Schaefer-Štefankovič Algorithm

Schaefer and Štefankovič proved the following theorem [25, Lemma 9].

► **Theorem 14.** *If G is a one-vertex non-orientable (respectively orientable) scheme (ρ, λ) , then it admits a cross-cap drawing with $eg(G)$ (respectively $eg(G) + 1$) cross-caps in which every edge passes through every cross-cap at most twice.*

Schaefer and Štefankovič gave an inductive algorithm computing the cross-cap drawing claimed by this theorem. We first introduce the different moves that they use to deal with different types of loops.

For an embedding scheme around a vertex v , by *flipping* a wedge of a one-sided loop e in a one-vertex scheme, we mean reversing the order of the edges in the wedge and changing the signature of the loops that have exactly one end in the wedge. We call an empty wedge between two consecutive half-edges around v a *root wedge*.

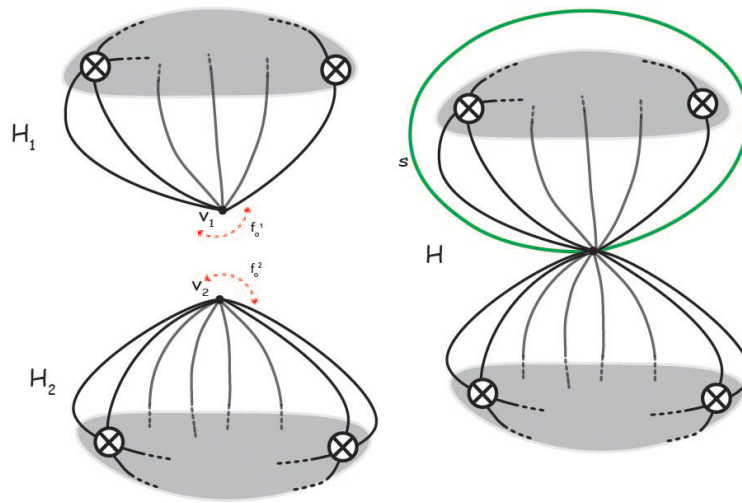
Contractible loop move. Let c be a contractible loop with consecutive ends in the scheme G . Remove c . The new scheme can be drawn using the same number of cross-caps. Having a drawing for the new scheme, we can draw the loop c without passing through any of the cross-caps.

Gluing move. Let s be a non-contractible separating loop in the scheme G . We divide the scheme to G_1 and G_2 by cutting along s and splitting the vertex into two vertices (the embedding schemes of G_1 and G_2 are induced by the embedding scheme of G). Denote by f_o^1 and f_o^2 the root wedges in G_1 and G_2 in which s was formerly placed. Let H_1 and H_2 be drawings for G_1 and G_2 respectively. We glue the drawings by identifying the root wedges f_o^1 and f_o^2 to get the drawing H' for $G \setminus \{s\}$.

Note that removing s does not change the genus and we have $eg(G) = eg(G_1) + eg(G_2)$. If G_1 and G_2 are both non-orientable, then H' can be extended to a cross-cap drawing for G by adding s without using any of the cross-caps; see Figure 5. When at least one of G_1 or G_2 is orientable, say G_2 , H' uses one extra cross-cap (G_2 needs $eg(G_2) + 1$ cross-caps to be drawn). To deal with this case, we need the following lemma and the dragging move which allows us to reduce one cross-cap from the drawing.

► **Lemma 15.** *Let (ρ, λ) be an orientable embedding scheme for the one vertex graph G . Adding a one-sided loop o with consecutive ends to the scheme (anywhere in the rotation around the vertex) increases the Euler genus by 1. Thus, the new scheme needs as many cross-caps as G to embed. Furthermore, the loop o is orienting.*

² Our Theorem 2 is stated with respect to the numbers of edges $|E(G_i)|$ for simplicity, but the proof also provides a bound in terms of the Betti numbers $\beta(G_i)$, as in Theorem 10.



■ **Figure 5** The gluing move on two cross-cap drawings when G_1 and G_2 are both non-orientable.

Dragging move. Let us assume that G_2 is orientable. By Lemma 15, we can add a one-sided loop o with consecutive ends in the root wedge f_o^2 without increasing the number of cross-caps that we need to draw G_2 . The loop o is orienting and the new scheme needs $eg(G_2) + 1$ cross-caps to be drawn. Having a drawing for the new scheme, we can draw the loop s in the drawing for $G_2 + \{o\}$ as follows: we start the loop from one of the root wedges between o and another loop of G_2 , we draw s by following o through all the cross-caps, except that after coming out of the last cross-cap, we go back to the first one entered, and traverse all of the cross-caps again. At the end, we follow o back to the vertex; see Figure 6, left. We denote this drawing of $G_2 + \{o\} + \{s\}$ by H'_2 .

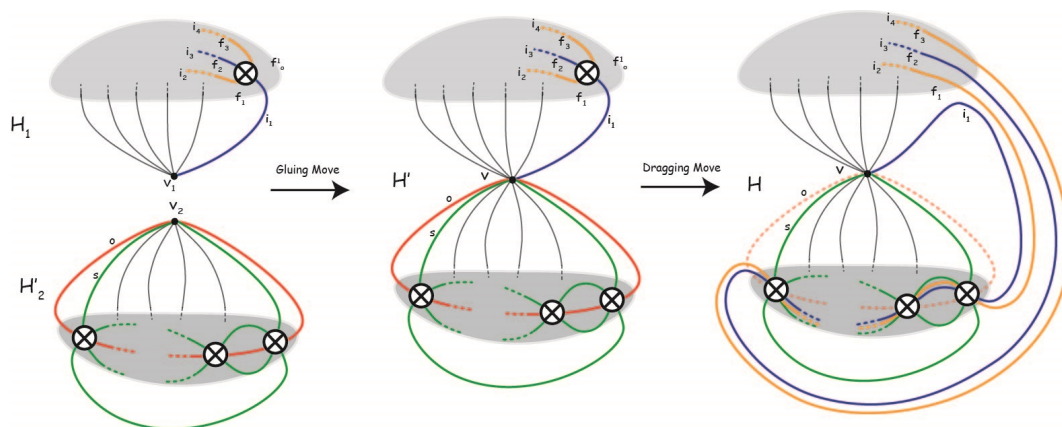
By gluing H_1 to H'_2 , we get a drawing H' for $G + \{o\} + \{s\}$ but the drawing is not using the minimum number of cross-caps. We eliminate one of the cross-caps in H' as follows.

Let i_1 be the rightmost half-edge in G_1 that follows immediately the separating loop in G . Denote by \mathfrak{c} the first cross-cap that i_1 passes through. Let us assume that there are $2k$ half-edges passing through \mathfrak{c} . Let us denote by $(i_1, f_1, \dots, i_{2k}, f_o^1)$ the alternating sequence of half-edges and faces adjacent to \mathfrak{c} in the cross-cap drawing by moving clockwise around it. Now, we disconnect the edges that enter \mathfrak{c} and remove the cross-cap \mathfrak{c} . We drag i_1, \dots, i_k through all the cross-caps in G_2 along the loop o . After exiting the last cross-cap in G_2 , we remove o and we attach the half edges to their other ends (i_{k+1}, \dots, i_{2k}) . Since G_2 uses an odd number of cross-caps (Lemma 7), the half edges will have the correct orientability and order to get attached to their other ends; see Figure 6. If only one of G_1 and G_2 is orientable, the drawing we get uses $eg(G)$ cross-caps and if both are orientable, we get a drawing with $eg(G) + 1$ cross-cap, that is, the minimum cross-cap needed to draw the scheme in this case.

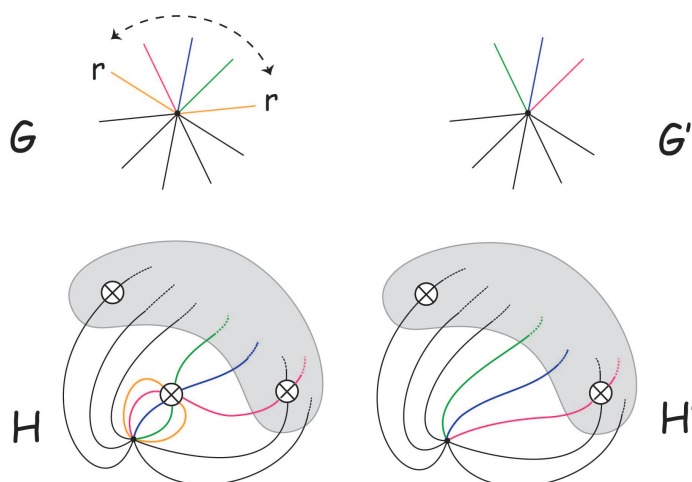
One-sided loop move. Let r be a one-sided loop in the scheme G . We remove r and flip one of its wedges. One can check that the new scheme G' has Euler genus $eg(G) - 1$. Let us assume that H' is a drawing for G' . We add r to this drawing by adding a cross-cap near the vertex and the flipped wedge and dragging r and every edge in the flipped wedge in it; see Figure 7. Note that flipping different wedges of r leads to two different cross-cap drawings. This freedom in choosing the wedge will be used later.

If we apply a one-sided loop move on an orienting loop, the drawing we get does not use the minimum amount of cross-caps, hence we need the following different move.

41:10 Short Topological Decompositions of Non-Orientable Surfaces



■ **Figure 6** Left: the gluing move. Right: the dragging move when G_2 is orientable: the top right crosscap is removed and the corresponding curves are dragged through the bottom component.

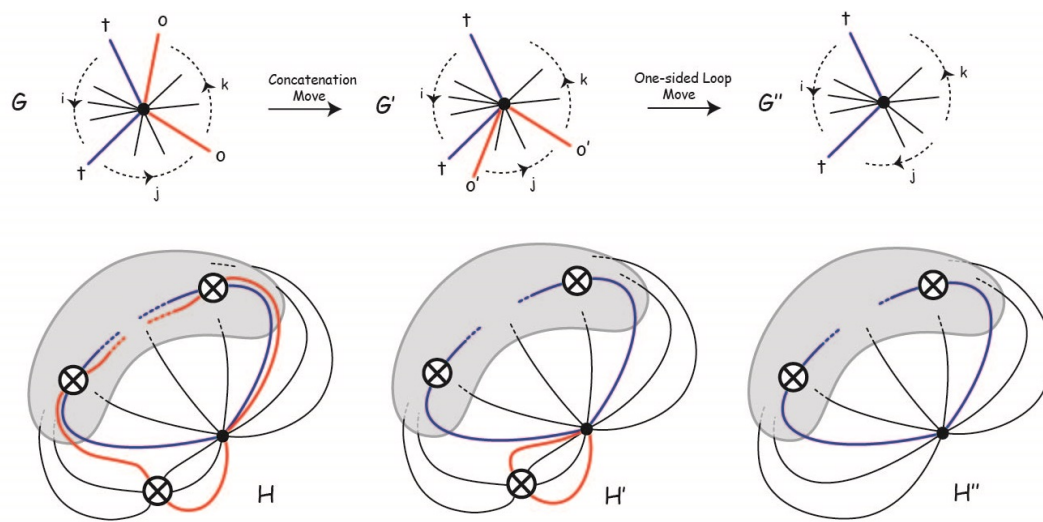


■ **Figure 7** The one-sided loop move on the loop r .

Concatenation move. Let o be an orienting loop in the scheme G such that one of its ends is immediately followed by an end of a two-sided non-separating loop t in the rotation. By Lemma 5, since t is non-separating, the concatenation of o and t which we denote by o' , is not orienting. Denote by G' the scheme in which we replace o by o' (we need $eg(G)$ cross-caps to draw both G and G'). If H' is a drawing for G' , one can obtain from H' a drawing of G by replacing the drawing o' by its concatenation with t . Depending on the wedge of o' that we choose to flip, we slide o' along t in the drawing:

If we flipped the wedge that does not encompass the loop t , we detach the end of o' next to t and slide it along t and we attach it to the vertex. This way, it ends up where the end of o was placed originally; see Figure 8.

If we flipped the wedge that encompasses the loop t , we draw o as follows: note that o' passes through only one cross-cap. We draw o next to the end of o' that is not slid along t , but instead of following o' into the cross-cap, we follow t . We can do this because the loop o' is next to the loop t in the rotation around this cross-cap; see Figure 9.



■ **Figure 8** The concatenation move when the flipped wedge does not encompass the ends of t .

The Schaefer-Štefankovič algorithm also uses an additional move which we will not need, so we do not introduce it here. Each of these moves provides a way to draw a loop assuming that some simpler one-vertex graph without that loop has already been drawn. The algorithm behind Theorem 14 works by applying these moves in a specific order, we refer to the full version for details. Note that by Lemma 6, in a drawing that is obtained by Theorem 14, every orienting loop passes through each cross-cap exactly once and if a separating loop enters a cross-cap, it passes through that cross-cap exactly twice.

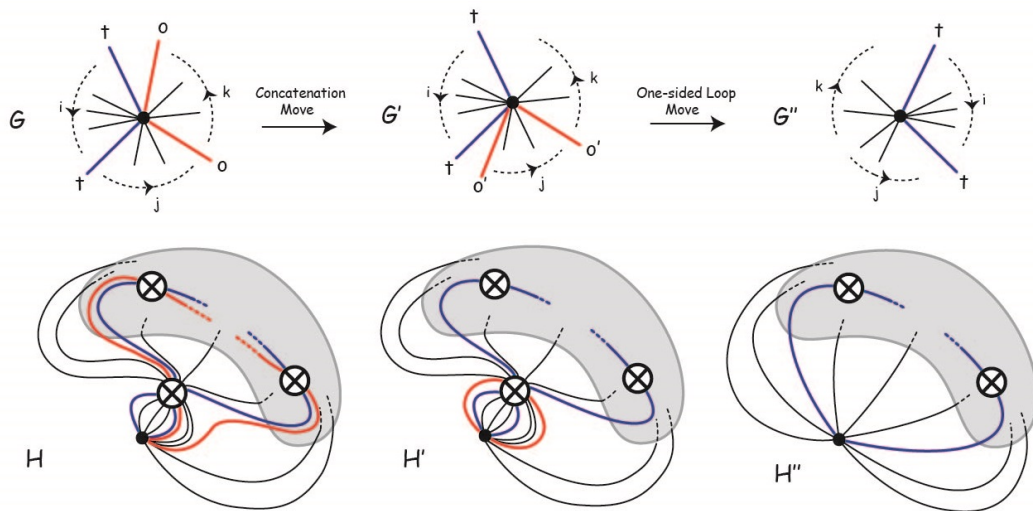
4.2 Our Modification to the Schaefer-Štefankovič algorithm

Our modification to the Schaefer-Štefankovič is to add two preprocessing steps, and then enforce more specific rules as to how to apply the moves described in the previous section. The first preprocessing is to add an orienting curve (via Lemma 9) and contract a spanning tree. In doing so, each edge in G is subdivided in at most 3 edges. This is summarized in the following lemma.

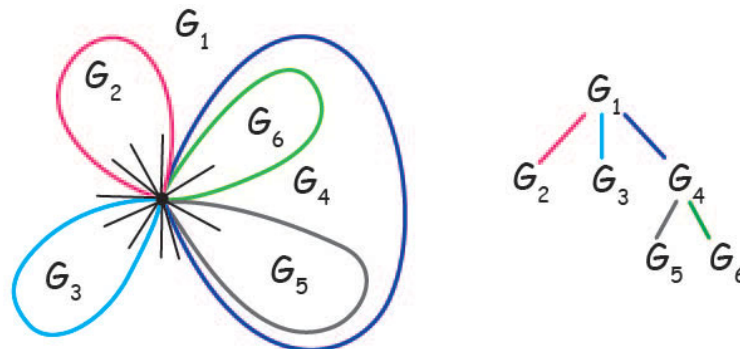
► **Lemma 16.** *Given a graph G embedded on a non-orientable surface N , there exists a one-vertex scheme \hat{G} such that \hat{G} has an orienting loop, and if \hat{G} has a non-orientable canonical system of loops of multiplicity at most k , then G has a non-orientable canonical system of loops of multiplicity at most $3k$.*

For the second preprocessing move we need a definition that is inspired by a similar notion from the literature on sorting signed permutations by reversals [12]. Given an embedding scheme G , the *interleaving graph* I_G has as vertex set the set of loops of G , and two vertices are connected if their corresponding loops have interleaving ends. When we talk about the sidedness of a vertex, we mean the sidedness of the loop it is associated to. A connected component in the interleaving graph is called *non-orientable* if it has a one-sided vertex, and *orientable* otherwise. A component with only one vertex is a *trivial* component, and *non-trivial* otherwise. Separating loops correspond to trivial orientable components.

41:12 Short Topological Decompositions of Non-Orientable Surfaces



■ **Figure 9** The concatenation move when the flipped wedge encompasses the ends of t .



■ **Figure 10** Left: a saturated one-vertex scheme in which the drawn loops are the separating loops; Right: the component tree of the left scheme. Note that the component G_4 is an empty sub-scheme.

Our second preprocessing step aims at subdividing G into subschemes G_i such that each I_{G_i} has only one non-trivial component. To achieve this, we *saturate* the scheme with auxiliary separating loops, i.e., we add a separating loop for any non-trivial component that is not divided from the rest of the graph by some separating loops. Since adding a separating loop does not interfere with the genus, then $eg(G) = eg(\bar{G})$ and we can later remove the added separating loops. Given a non-orientable scheme G saturated with separating loops and any cellular embedding of G on a surface N , cutting G along the separating loops yields subsurfaces N_i of N , each containing (possibly empty) components of G , which we denote by G_i (see Figure 10). The *component tree* of G has a vertex for every such sub-graph G_i , and two vertices are connected if their corresponding components are separated by a separating loop. See Figure 10 for an example of a component tree.

We now describe our algorithm. Throughout the main loop of our algorithm, if we have homotopic loops we remove all of them except one and after drawing this one, we re-introduce them parallel to the one drawn.

A difference of our modified version with the original one is that it is not clear at first sight that we cover all cases. This is justified by the following lemmas.

► **Lemma 17.** *A scheme with an orienting loop is non-orientable.*

■ **Algorithm 1** The modified algorithm.

Pre-processing steps:

- **Step A:** If there is no orienting loop, we add an orienting loop and contract a spanning tree using Lemma 16.
- **Step B:** If G is not saturated by separating curves, we saturate it.

Main loop:

- **Step 1: If there is a contractible loop.** We recurse on the scheme without the loop and apply the contractible loop move.
- **Step 2: If there exists a separating (non-contractible) loop.** We pick a separating loop that separates a non-root leaf from the component tree, recurse on the subschemes and apply a gluing and a dragging move.
- **Step 3.1: If there exists a one-sided non-orienting loop.** We pick a one-sided non-orienting loop such that the scheme G' that we obtain when removing it and flipping its wedge maximizes the number of one-sided loops. We recurse on G' and apply the one-sided loop move on this loop.
- **Step 3.2.a: If all one-sided loops are orienting and there are two-sided loops.** We pick an orienting loop adjacent to a two-sided loop, recurse on the drawing H' described in the concatenation move and apply the concatenation move on these loops.
- **Step 3.2.b: If all one-sided loops are orienting and there are no two-sided loops.** In this case one cross-cap is sufficient to draw all the loops.

Post-processing steps:

- **Step B':** Erase the extra separating loops added in step B.
 - **Step A':** Uncontract the spanning tree and remove the loop added in step A.
-

As a corollary, a scheme G that has an orienting loop needs $eg(G)$ cross-caps to be drawn.

► **Lemma 18.** *Let G be a scheme with an orienting loop o and a non-contractible separating loop s . Then s separates the graph into an orientable and a non-orientable sub-graph.*

Then the following lemma shows that there is always an orienting curve throughout the recursive calls of the algorithm. We state it as a corollary since it is a direct corollary of three technical lemmas that are featured in the full version.

► **Corollary 19.** *Let G be a one-vertex scheme with an orienting loop. Let G' be the graph on which the modified algorithm recurses when applying a contractible loop move, a one-sided loop move or a concatenation move. Then G' has an orienting loop. Likewise, when the modified algorithm applies a gluing and dragging move on a separating loop s , the two subgraphs G_1 and G_2 on which it recurses have an orienting loop.*

We now analyze the cross-cap drawing provided by the modified algorithm.

► **Lemma 20.** *Let G be a graph cellularly embedded on a non-orientable surface. If G has an orienting loop, applying the modified algorithm, we obtain a cross-cap drawing of G with $eg(G)$ cross-caps such that each loop of G enters each cross-cap at most twice. Otherwise, we obtain a cross-cap drawing of G with $eg(G)$ cross-caps such that each loop of G enters each cross-cap at most 6 times.*

Sketch of proof. After the preprocessing steps, we obtain a saturated scheme with an orienting loop and by Lemma 16 we can work with this scheme to prove the lemma. We follow the recursive steps of the modified algorithm, and thus provide a proof by induction on $eg(G) + |E(G)|$. By Corollary 19, there is an orienting loop throughout the recursive calls of the algorithm. Based on the description of each move, it can be seen that at each step we obtain a drawing with a minimum number of cross-caps. It is easy to see that the induction carries over after Step 1. Then, by Lemma 18, we know that a separating loop divides the scheme into a non-orientable and an orientable subscheme. This case is handled by Step 2 of the algorithm. One can see that in the dragging move and drawing the separating loop, each edge follows the auxiliary orienting loop o added to G_2 at most twice. Since o passes through every crosscap in the drawing of G_2 exactly once, we get the right multiplicity for every edge. For Step 3.1, let G' be the scheme that we recurse on. By the recursion of the algorithm we obtain a drawing for G' in which each edge uses each cross-cap at most twice. In adding the last cross-cap in the one-sided loop move, only the half-edges in the flipped wedge enter the last cross-cap and therefore each edge enters this cross-cap at most twice. For Step 3.2.a, let o' be the concatenation of the loops o and t . The loop o' is the only one-sided non-orienting loop in G' . By applying a one-sided loop move on o' and then recursing, we obtain a drawing for G' in which o' and t pass through a disjoint set of cross-caps and at most twice through each. Then it can be shown that sliding o' back along t , o' enters every cross-cap at most twice. Step 3.2.b is a base case for the induction satisfying the requirements of the lemma. Finally, since there is always an orienting loop, by Lemma 17 there is always at least one one-sided loop, and thus we are never in a case not covered by the previous steps. This concludes this sketch. ◀

The following two lemmas ensure further properties guaranteed by our algorithm and explain our choice for the rule in Step 3.1. The proof of the second lemma mirrors proofs in the signed reversal distance theory [2, Theorem 1].

► **Lemma 21.** *If there exists an orienting loop o in the embedding scheme G , the connected component that has the vertex o is the only non-orientable component in I_G .*

► **Lemma 22.** *Let G be a one-vertex scheme with an orienting loop and no non-contractible separating loop such that I_G has only one non-trivial component. Then G can be drawn by exclusively applying a sequence of contractible, concatenation one-sided loop moves.*

4.3 The Non-orientable Canonical System of Loops

Our algorithm has the following key advantage compared to the algorithm of Schaefer and Štefankovič: due to the order in which we choose the loops in Steps 2 and 3.1, we know that dragging moves and the other moves do not intermingle during the recursive calls of the algorithm. Indeed, first, by Lemma 22, when it draws a scheme with a single non-trivial component, it only relies on contractible, concatenation and one-sided loop moves. Second, due to the order in which we choose the loops in Steps 2 and 3.1 and Lemma 21, we know that whenever a dragging move is applied, the orientable sub-scheme on which we recurse has only one non-trivial component. In this section, we leverage these two key advantages to find a non-orientable canonical system of loops of small multiplicity.

A *root face* in a cross-cap drawing is a face adjacent to the vertex. The strategy to prove Theorem 3 is to show that the modified algorithm outputs a cross-cap drawing where cross-caps are not too far from the vertex.

► **Lemma 23.** *For any saturated one-vertex scheme G with an orienting loop o , the cross-cap drawing H output by the modified algorithm has $eg(G)$ cross-caps, and there is a path from every cross-cap to a root face (not necessarily fixed) with multiplicity at most two.*

This lemma is the crux of the paper. We refer to the full version for the proof, and only outline some ideas. The difficulty lies in the fact that long dual paths are added to the cross-cap drawings when doing one-sided loop moves and dragging moves, making it hard to control the diameter of the graph dual to the cross-cap drawings throughout the recursive calls. This is solved by tracking specific paths, whose lengths are controlled using two different strategies for the one-sided loop moves and for the dragging moves. The inductive property that we maintain throughout one-sided loop moves is that there is a short path *that does not cross the orienting curve* from every crosscap to a *fixed* root face. The choice of a fixed root face ensures that, if one chooses carefully the wedge when applying a one-sided loop move, the lengths of the paths stay controlled. The property of not crossing the orienting curve is key because, when we subsequently apply dragging moves, edges dual to the orienting curves might get replaced with long dual paths (see Figure 6). However, dragging moves naturally lead to paths that connect different root faces. Fortunately, at this stage, this is not an issue since all the one-sided moves have already been applied. Therefore, for this proof strategy to succeed, our modifications are crucial, in particular to ensure that one-sided loop moves and dragging moves do not alternate during recursive calls.

Finally, we now have all the tools to prove Theorem 3.

Sketch of proof of Theorem 3. The modified algorithm on G constructs a cross-cap drawing of a saturated scheme with an orienting loop. By Lemma 20, the loops enter each cross-cap at most twice. By Lemma 23, we furthermore have paths $\{p_j\}$ of multiplicity two from a face incident to each cross-cap to a root face in this cross-cap drawing. We follow these paths to construct loops surrounding each of the cross caps. By Lemma 8, the system of loops we obtain is canonical. Adding the different bounds on the multiplicity, we obtain that each loop in the system has multiplicity 10. ◀

References

- 1 Dan Archdeacon and C Paul Bonnington. Two maps on one surface. *Journal of Graph Theory*, 36(4):198–216, 2001.
- 2 Anne Bergeron. A very elementary presentation of the Hannenhalli-Pevzner theory. In *Annual Symposium on Combinatorial Pattern Matching*, pages 106–117. Springer, 2001.
- 3 Andrei C Bura, Ricky XF Chen, and Christian M Reidys. On a lower bound for sorting signed permutations by reversals. *arXiv preprint*, 2016. [arXiv:1602.00778](https://arxiv.org/abs/1602.00778).
- 4 Éric Colin de Verdière. Topological algorithms for graphs on surfaces. Habilitation thesis, <http://www.di.ens.fr/~colin/>, 2012.
- 5 Éric Colin de Verdière. Computational topology of graphs on surfaces. In Jacob E. Goodman, Joseph O’Rourke, and Csaba Toth, editors, *Handbook of Discrete and Computational Geometry*, chapter 23, pages 605–636. CRC Press LLC, third edition, 2018.
- 6 Éric Colin De Verdière and Jeff Erickson. Tightening nonsimple paths and cycles on surfaces. *SIAM Journal on Computing*, 39(8):3784–3813, 2010.
- 7 Jeff Erickson and Sarel Har-Peled. Optimally cutting a surface into a disk. *Discrete & Computational Geometry*, 31(1):37–59, 2004.
- 8 Jeff Erickson and Kim Whittlesey. Greedy optimal homotopy and homology generators. In *SODA*, volume 5, pages 1038–1046, 2005.
- 9 Niloufar Fuladi, Alfredo Hubard, and Arnaud de Mesmay. Short topological decompositions of non-orientable surfaces, 2022. [arXiv:2203.06659](https://arxiv.org/abs/2203.06659).

- 10 Jim Geelen, Tony Huynh, and R Bruce Richter. Explicit bounds for graph minors. *Journal of Combinatorial Theory, Series B*, 132:80–106, 2018.
- 11 Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.
- 12 Sridhar Hannenhalli and Pavel A Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM (JACM)*, 46(1):1–27, 1999.
- 13 Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- 14 Brian Hayes. Computing science: Sorting out the genome. *American Scientist*, 95(5):386–391, 2007.
- 15 Petr Hliněný and Gelasio Salazar. On hardness of the joint crossing number. In *International Symposium on Algorithms and Computation*, pages 603–613. Springer, 2015.
- 16 Fenix WD Huang and Christian M Reidys. A topological framework for signed permutations. *Discrete Mathematics*, 340(9):2161–2182, 2017.
- 17 Francis Lazarus. Combinatorial graphs and surfaces from the computational and topological viewpoint followed by some notes on the isometric embedding of the square flat torus. *Mémoire d'HDR*, 2014. Available at <http://www.gipsa-lab.grenoble-inp.fr/~francis.lazarus/Documents/hdr-Lazarus.pdf>.
- 18 Francis Lazarus, Michel Pocchiola, Gert Vegter, and Anne Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proceedings of the seventeenth annual symposium on Computational geometry*, pages 80–89, 2001.
- 19 Jiří Matoušek, Eric Sedgwick, Martin Tancer, and Uli Wagner. Untangling two systems of noncrossing curves. In *International Symposium on Graph Drawing*, pages 472–483. Springer, 2013.
- 20 Bojan Mohar. The genus crossing number. *ARS Mathematica Contemporanea*, 2(2):157–162, 2009.
- 21 Bojan Mohar and Carsten Thomassen. *Graphs on surfaces*, volume 10. JHU press, 2001.
- 22 Seiya Negami. Crossing numbers of graph embedding pairs on closed surfaces. *Journal of Graph Theory*, 36(1):8–23, 2001.
- 23 R Bruce Richter and Gelasio Salazar. Two maps with large representativity on one surface. *Journal of Graph Theory*, 50(3):234–245, 2005.
- 24 Marcus Schaefer and Daniel Štefankovič. Block additivity of \mathbb{Z}_2 -embeddings. In *International Symposium on Graph Drawing*, pages 185–195. Springer, 2013.
- 25 Marcus Schaefer and Daniel Štefankovič. The degenerate crossing number and higher-genus embeddings. *Journal of Graph Algorithms and Applications*, 26(1):35–58, 2022. doi:10.7155/jgaa.00580.
- 26 James P Sethna. Order parameters, broken symmetry, and topology. In *1991 Lectures in Complex Systems*. Addison-Wesley, 1992.
- 27 Alla Sheffer, K Hormann, B Levy, M Desbrun, K Zhou, E Praun, and H Hoppe. Mesh parameterization: Theory and practice. *ACM SIGGRAPH, course notes*, 10(1281500.1281510), 2007.
- 28 John Stillwell. *Classical topology and combinatorial group theory*, volume 72. Springer Science & Business Media, 1993.

Robust Radical Sylvester-Gallai Theorem for Quadratics

Abhibhav Garg  

Cheriton School of Computer Science, University of Waterloo, Canada

Rafael Oliveira¹  

Cheriton School of Computer Science, University of Waterloo, Canada

Akash Kumar Sengupta 

Department of Mathematics, Columbia University, New York, NY, USA

Abstract

We prove a robust generalization of a Sylvester-Gallai type theorem for quadratic polynomials. More precisely, given a parameter $0 < \delta \leq 1$ and a finite collection \mathcal{F} of irreducible and pairwise independent polynomials of degree at most 2, we say that \mathcal{F} is a $(\delta, 2)$ -radical Sylvester-Gallai configuration if for any polynomial $F_i \in \mathcal{F}$, there exist $\delta(|\mathcal{F}| - 1)$ polynomials F_j such that $|\text{rad}(F_i, F_j) \cap \mathcal{F}| \geq 3$, that is, the radical of F_i, F_j contains a third polynomial in the set.

We prove that any $(\delta, 2)$ -radical Sylvester-Gallai configuration \mathcal{F} must be of low dimension: that is

$$\dim \text{span}_{\mathbb{C}} \{ \mathcal{F} \} = \text{poly}(1/\delta).$$

2012 ACM Subject Classification Theory of computation \rightarrow Algebraic complexity theory; Theory of computation \rightarrow Computational geometry

Keywords and phrases Sylvester-Gallai theorem, arrangements of hypersurfaces, locally correctable codes, algebraic complexity, polynomial identity testing, algebraic geometry, commutative algebra

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.42

Related Version *Full Version:* <https://arxiv.org/abs/2203.05532>

Independent result

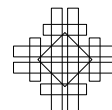
We would like to remark that, independently and simultaneously to our work, [21] have also proved that $(\delta, 2)$ -radical Sylvester-Gallai configurations must be of low dimension. Both works have been presented in a common talk at CG week 2022. For a more detailed comparison between both works, we refer the reader to Section 1.3.

1 Introduction

Suppose $v_1, \dots, v_m \in \mathbb{R}^n$ is a set of m distinct points, such that the line joining any two points in the set contains a third point. In 1893, Sylvester asked if such configurations of points are necessarily colinear [26]. Independently, this same question was asked by Erdős in 1943 [9]. This was proved by [17], and independently by Gallai [10], where the latter was in response to Erdős. This result is now known as the Sylvester-Gallai theorem. A set of points satisfying the above is called a Sylvester-Gallai (SG) configuration.

Sylvester-Gallai theorems depend on the base field. For instance, it is well known that any nonsingular planar cubic curve over \mathbb{C} has nine inflection points, and that any line passing through two such points passes through a third [5]. These nine points are not collinear,

¹ corresponding author



and therefore form a counterexample to the Sylvester-Gallai theorem when the underlying field is changed from \mathbb{R} to \mathbb{C} . In 1966, Serre asked if there are configuration of points in \mathbb{C}^n that satisfy the Sylvester-Gallai that are not coplanar [23]. Kelly [15] proved that no such configurations can exist, or equivalently that points in \mathbb{C}^n that satisfy the Sylvester-Gallai property are always coplanar.

Over finite fields, Sylvester-Gallai configurations do not have bounded dimension. For example, if we are working over the field \mathbb{F}_p (with $p > 2$) and our vector space is \mathbb{F}_p^n , then the set of points is \mathbb{F}_p^n is a SG configuration of dimension n , which is not constant. In general, any subgroup of \mathbb{F}_p^n will form a Sylvester-Gallai configuration. Some bounds on the dimension of configurations in this setting can be found in [7]. In this work, we only focus on fields of characteristic zero, and to make presentation easier we restrict our attention to \mathbb{C} .

Several variations and generalizations of the Sylvester-Gallai problem defined above have been studied in combinatorial geometry. The underlying theme in all these types of questions is the following:

Are Sylvester-Gallai type configurations always low-dimensional?

In characteristic zero, the answer has always turned out to be yes. For a thorough survey of the earlier works on SG-type theorems, we refer the reader to [2] and results therein.

While the above results are mathematically beautiful and interesting on their own right, it is also interesting and useful in areas such as computer science and coding theory to consider higher-dimensional analogs as well as robust analogs of SG type theorems.

Higher-dimensional analogs of SG configurations. In [12], a higher dimensional version of the theorem was proved, with lines replaced by flats. This variant has applications in the study of algebraic circuits, and in particular in Polynomial Identity Testing (PIT) [14, 22], a central problem in algebraic complexity theory. The works [14, 22] use the higher dimensional Sylvester-Gallai theorems to bound the “rank” of certain types of depth three circuits.² In simple terms, if the linear forms of a circuit satisfy the high dimensional SG condition, then in essence the polynomial being computed must depend on a constant number of variables, in which case it is easy to check whether the circuit is computing a non-zero polynomial.

Robust analogs of SG configurations and applications. Robust generalizations of the Sylvester-Gallai theorem have found applications in coding theory and in complexity theory.

In this variant, for every point v_i there are at least $\delta(m - 1)$ points u_1, \dots, u_k such that v_i and u_k span a third point. The usual Sylvester-Gallai theorem is the case when $\delta = 1$. Such configurations were first studied by Szemerédi and Trotter [27], who proved that if δ is bigger than an absolute constant close to 1, then the configuration has constant dimension.

In [1], the authors prove that such a configuration has dimension $\mathcal{O}(1/\delta^2)$, for any $0 < \delta \leq 1$. This robust version also allows them to prove robust versions of the higher dimensional variants, and average case versions of the theorem. They also define the notion of a *LCC*-configuration, which is an extension of the Sylvester-Gallai configuration where points are allowed to occur with multiplicity. In [8], the authors improve the bound on the dimension of robust Sylvester-Gallai configurations to $\mathcal{O}(1/\delta)$.

In coding theory, these robust configurations naturally appear in the study of locally decodable codes and locally correctable codes [1]. These results, as well as similar results are surveyed in [7]. They also have applications in the study of algebraic circuits, in particular in reconstruction of algebraic circuits [25].

² Algebraic circuits which compute polynomials that can be written as a sum of products of linear forms.

Higher degree generalizations of Sylvester-Gallai configurations. Also motivated by the PIT problem, Gupta in [11] introduced higher degree analogs of Sylvester-Gallai configurations, and asked if they are also “low dimensional.” In his paper, Gupta outlines a series of conjectures, and gives a deterministic polynomial-time blackbox PIT algorithm for a special class of algebraic circuits³ assuming that these conjectures hold.

The first challenge in Gupta’s series of conjectures on SG type theorems is the following:

► **Conjecture 1** (Conjecture 29, [11]). *Let $Q_1, \dots, Q_m \in \mathbb{C}[x_1, \dots, x_n]$ be irreducible, homogeneous, and of degree at most d such that for every pair Q_i, Q_j there is a k such that $Q_k \in \text{rad}(Q_i, Q_j)$. Then the transcendence degree of Q_1, \dots, Q_m is $\mathcal{O}(1)$ (where the constant depends on the degree d).*

The case of $d = 2$ for the above conjecture was proved in [24]. We henceforth refer to the original Sylvester-Gallai theorem (the case $d = 1$) and its variants as the “linear case”. As in the linear case of SG type problems, it is natural to consider the robust version of the above lemma as a next step towards the conjectures of Gupta that give an algorithm for a special case of PIT. We resolve the robust version of the above conjecture in the case when $d = 2$.

1.1 Main results

In this section, we formally state our main result: robust quadratic radical Sylvester-Gallai configurations must lie in a constant dimensional vector space.⁴ In particular, this result implies that the polynomials must be contained in a small algebra, and also that they have constant transcendence degree. Another important result is a structural result for ideals generated by two quadratic forms.

1.1.1 Robust radical Sylvester-Gallai theorem

We first formally define robust quadratic radical Sylvester-Gallai configurations. Henceforth, as is customary in the literature, we will use *form* to denote homogeneous polynomials. For a polynomial ring $S = \mathbb{C}[x_1, \dots, x_n]$, we let S_d denote the vector space of polynomials of degree d in S , and the ideal generated by a set of polynomials f_1, \dots, f_r is denoted as (f_1, \dots, f_r) . We also use $\text{rad}(f_1, \dots, f_r)$ to denote the radical of this ideal, that is, the set of polynomials g such that $g^k \in (f_1, \dots, f_r)$ for some k .

► **Definition 2** ($(\delta, 2)$ -rad-SG configurations). *Let $0 < \delta \leq 1$ and $\mathcal{F} := \{F_1, \dots, F_m\}$ be a set of irreducible forms in the polynomial ring $S = \mathbb{C}[x_1, \dots, x_n]$. We say that \mathcal{F} is a $(\delta, 2)$ -rad-SG configuration if the following conditions hold:*

1. $\mathcal{F} \subset S_1 \cup S_2$ (only linear and quadratic forms)
2. for any $i \neq j$, we have that $F_i \notin (F_j)$
3. for any $i \in [m]$, there are $\delta(m - 1)$ indices $j \in [m] \setminus \{i\}$ such that $|\text{rad}(F_i, F_j) \cap \mathcal{F}| \geq 3$.

We are now ready to formally state the main contributions of our paper. We begin with our main theorem, that robust quadratic radical SG configurations must have small linear span.

► **Theorem 3** ($(\delta, 2)$ -rad-SG theorem). *If \mathcal{F} is a $(\delta, 2)$ -rad-SG configuration, then*

$$\dim(\text{span}_{\mathbb{C}} \{\mathcal{F}\}) = O(1/\delta^{54}).$$

³ These are circuits computed by a sum of constantly many products of constant degree polynomials.

⁴ Our results hold for any algebraically closed field of characteristic zero. However, for simplicity of exposition, we only state our results over \mathbb{C} .

To prove the theorem above, we first notice that the theorem would imply that the forms in the configuration are contained in a subalgebra of the polynomial ring of small dimension, namely the subalgebra generated by a linear basis of the given configuration. With this observation at hand, we provide a principled approach to construct small dimensional subalgebras of the polynomial ring which control the configuration (in the sense that all forms in the configuration will become a “univariate form” with coefficients from our subalgebra).

The main property of these algebras is that they allow us to translate non-linear SG dependencies (the radical dependencies) into linear SG dependencies, and therefore we can reduce our non-linear problem to the linear version of the SG problem.

The main principle guiding the construction of our subalgebras is that we would like these subalgebras to look “as free as possible” without increasing the dimension of the algebra by much. The amount of “freeness” that we need is captured by the robust algebras defined in Section 2, where we also elaborate on how these algebras behave with SG configurations (where we need the notion of clean algebras). For more intuition on how we prove the theorem, we refer the reader to Section 1.2.

1.1.2 Results on structure of ideals generated by two quadratics

A key step in our strategy to prove that a SG configuration is low dimensional (as has also been the first step in the works of [24, 19]) is to understand the structure of ideals generated by two quadratic forms.

The general principle at play here is that if the ideal generated by two quadratic forms is not radical or prime, then there must be a low-rank quadratic in their span. In [24, 19], the authors proved similar structural results to determine when a product of quadratic forms is contained in an ideal generated by two quadratic forms. In Proposition 4, we use a different approach to completely characterize when the ideal generated by two quadratic forms is radical or prime, and as corollaries we obtain the structural results in [24, 19]. We use a commutative-algebraic approach to develop a further understanding of the radical of ideals generated by two irreducible quadratics. Indeed, using the standard tools of primary decomposition and Hilbert-Samuel multiplicity we obtain a classification of the possible minimal primes of an ideal generated by two quadratic forms. Consequently we obtain a characterization for such an ideal to be prime or radical. This approach can also be generalized to ideals generated by cubic forms ([18]).

► **Proposition 4 (Radical Structure).** *Let \mathbb{K} be an algebraically closed field of characteristic zero and $Q_1, Q_2 \in S = \mathbb{K}[x_1, \dots, x_n]$ be two forms of degree 2. Then one of the following holds:*

1. *The ideal (Q_1, Q_2) is prime.*
2. *The ideal (Q_1, Q_2) is radical, but not prime. Furthermore, one of the following cases occur:*
 - (a) *There exist two linearly independent linear forms $x, y \in S_1$ such that $xy \in \text{span}(Q_1, Q_2)$.*
 - (b) *There exists a minimal prime $\mathfrak{p} \supset (Q_1, Q_2)$, such that $\mathfrak{p} = (x, y)$ for some linearly independent forms $x, y \in S_1$*
3. *The ideal (Q_1, Q_2) is not radical and one of the following cases occur:*
 - (a) *Q_1, Q_2 have a common factor and $Q_1 = xy$, $Q_2 = x(\alpha x + \beta y)$ for some linear forms x, y and $\alpha, \beta \in k$. In this case, we have $x^2 \in \text{span}(Q_1, Q_2)$.*
 - (b) *Q_1, Q_2 do not have a common factor. There exists a minimal prime $\mathfrak{p} \supset (Q_1, Q_2)$ such that $\mathfrak{p} = (x, Q)$, where $x \in S_1$, $Q \in S_2$ and Q is irreducible modulo x , and we also have $x^2 \in \text{span}(Q_1, Q_2)$.*

- (c) Q_1, Q_2 do not have a common factor and there exists a minimal prime $\mathfrak{p} \supset (Q_1, Q_2)$, such that $\mathfrak{p} = (x, y)$ for some linearly independent forms $x, y \in S_1$, and the (x, y) -primary ideal \mathfrak{q} has multiplicity $e(S/\mathfrak{q}) \geq 2$.

The proposition above is not new, and proofs of some of the statements can be found in [4, Section 1] and [13, Chapter XIII]. For completeness, we provide a proof of this proposition using primary decomposition and Hilbert-Samuel multiplicity of an ideal. In the former, the authors study the cycle decomposition of the intersection of two quadric hypersurfaces to obtain results about existence of rational points on intersection of two quadric hypersurfaces and Châtelet surfaces over number fields. Our statements here are slightly simpler to state (and slightly different) since in the works above the authors work in the more general setting of perfect fields, and we are only concerned with algebraically closed fields of characteristic zero.

1.2 Sketch of the proof of Theorem 3

In this section, we give a sketch of the proof of our main theorem. Suppose we are given a $(\delta, 2)$ -rad-SG configuration $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$, where \mathcal{F}_d is the set of forms of degree d in our configuration. We will show that there is a small subalgebra of the polynomial ring that contains \mathcal{F} . That is, we will construct a subalgebra $\mathbb{C}[y_1, \dots, y_s, Q_1, \dots, Q_t]$ generated by linear forms y_i and quadratic forms Q_j , such that $\mathcal{F} \subset \mathbb{C}[y_1, \dots, y_s, Q_1, \dots, Q_t]$ and $s + t = O(1/\delta^{27})$. Then it will follow that $\dim(\text{span}_{\mathbb{C}}\{\mathcal{F}\})$ is at most $O(1/\delta^{54})$, since every quadratic form in this algebra is a linear combination of the forms Q_j and pairwise products of the forms y_i .

A motivating special case. Our strategy to prove the robust radical SG theorem is based on the following toy example. Suppose our polynomial ring is $\mathbb{C}[x_1, \dots, x_r, y_1, \dots, y_s]$, where one should think of s being constant and $r \gg s$, and every quadratic form Q in our configuration is a polynomial which is “univariate” over the smaller polynomial ring $\mathbb{C}[y_1, \dots, y_s]$. That is, for each quadratic Q , there exists a linear form $x_Q \in \text{span}_{\mathbb{C}}\{x_1, \dots, x_r\}$ such that $Q \in \mathbb{C}[x_Q, y_1, \dots, y_s]$. In this case, one would hope that the *non-linear* SG dependencies involving our configuration \mathcal{F} would imply *linear* SG dependencies for the set of linear forms $\mathcal{F}_1 \cup \{x_Q \mid Q \in \mathcal{F}_2\}$. If we manage to prove that the latter set of linear forms is a *robust linear SG configuration*, we can invoke the robust SG theorem for linear forms of [1, 8] to bound the dimension of $\text{span}_{\mathbb{C}}\{\mathcal{F}_1 \cup \{x_Q \mid Q \in \mathcal{F}_2\}\}$. Thus we may take our small subalgebra to be the subalgebra generated by y_1, \dots, y_s and $\mathcal{F}_1 \cup \{x_Q \mid Q \in \mathcal{F}_2\}$.

Small subalgebras. In general it is not always possible to reduce the general robust radical SG problem for quadratics to the toy example above.⁵ However we will be able to construct a small subalgebra of our polynomial ring which is just as good as the small polynomial ring $\mathbb{C}[y_1, \dots, y_s]$ in the toy example above. Additionally, we will not always be able to reduce the non-linear problem to a robust linear SG configuration. Instead of a robust linear SG configuration, we will reduce it to a δ -LCC configuration of [1].

Since the main counterexample to the above toy example are quadratics of large rank, the small subalgebras that we construct will have both linear and quadratic forms as generating elements. Therefore, it is natural to consider the vector space of forms generating the algebra,

⁵ For instance if the polynomial $Q = \sum_{i=1}^s x_i y_i$ is in our SG configuration.

which we denote by $V := V_1 + V_2$, where V_1 is the vector space of linear forms in the algebra and V_2 is the vector space of quadratic *generators* of the algebra. The main idea here is that the quadratic generators will be composed only of quadratics of high rank, which can essentially be thought of as “free variables.” As it turns out, intuitively and informally, the only properties that we need from the vector space above are that:

1. the quadratics in V_2 are “robust” against the linear forms in V_1 . That is, we would like each quadratic in V_2 to be of very high rank even if we subtract from it polynomials from the algebra $\mathbb{C}[V_1]$
2. V is in a sense “saturated” with respect to our configuration \mathcal{F} . That is, there exists no small vector space of linear forms that we can add to V_1 that would add many polynomials of \mathcal{F} to the algebra $\mathbb{C}[V]$, or “make them closer to being in $\mathbb{C}[V]$.”

The first condition ensures that any quadratic from our set \mathcal{F} which “depends” on a form from V_2 must be of high rank, while the second condition ensures that there is no trivial way to increase the algebra slightly in order to have a lot more forms from \mathcal{F} inside of the larger algebra. We call any vector space which satisfies the conditions above a *clean vector space* with respect to \mathcal{F} . The formal definition of these vector spaces and the results needed can be found on Section 2.

Univariate over an intermediate small subalgebra. First we construct an intermediate small subalgebra $\mathbb{C}[V]$ such that any polynomial in \mathcal{F} is either contained in $\mathbb{C}[V]$ or it is univariate over $\mathbb{C}[V]$. To construct the subalgebra above, we need to understand in a bit more detail the structure of the radical of an ideal generated by two quadratic forms. To this end, we prove Proposition 4, generalizing the previous structure theorems from [24, 19]. Additionally, we also assemble results on the structure of minimal primes of these ideals to construct our algebra.

With Proposition 4 (our main structural result) at hand, we proceed similarly to [24, 19] by partitioning the quadratics in our $(\delta, 2)$ -rad-SG configuration \mathcal{F} into four subsets, each satisfying a particular case of our structure theorem. Taking $\varepsilon = \delta/10$, we define

1. $\mathcal{F}_{\text{span}}$ is the set of quadratics Q which satisfy a “span dependency” with at least ε -fraction of the polynomials. That is, there exist many quadratics $F, G \in \mathcal{F}$ such that $G \in \text{span}_{\mathbb{C}}\{Q, F\}$.
2. $\mathcal{F}_{\text{linear}}$ is the set of quadratics Q which satisfy case 3 (c) of Proposition 4 with at least an ε -fraction of the other polynomials. That is, there are many quadratics $F \in \mathcal{F}$ and linear forms x, y such that $(Q, F) \subset (x, y)$, and this minimal prime has multiplicity ≥ 2 .
3. \mathcal{F}_{deg} is the set of quadratics Q which have an ε -fraction of its SG dependencies with linear forms.⁶
4. $\mathcal{F}_{\text{square}}$ is the set of quadratics Q which satisfy case 3 (b) of Proposition 4 with at least $(\delta - 3\varepsilon)$ -fraction of the other polynomials. That is, there are many quadratics $F \in \mathcal{F}$ such that there is a linear form ℓ such that $\ell^2 \in \text{span}_{\mathbb{C}}\{F, Q\}$.

With this partition, we construct a small clean vector space V such that $\mathcal{F}_{\text{square}}$ and $\mathcal{F}_{\text{linear}}$ are entirely contained in the algebra $\mathbb{C}[V]$, and the forms in the remaining subsets are either in $\mathbb{C}[V]$, or are univariate over $\mathbb{C}[V]$. Here, by univariate over $\mathbb{C}[V]$, we mean that there is a linear form $z \notin V_1$ such that the polynomial is in the algebra $\mathbb{C}[V][z]$.

⁶ Deg stands for the degenerate case.

Construction of the intermediate small subalgebra. We construct the subalgebra above in four steps, where in each step we construct intermediate subalgebras which handle one of the subsets of quadratics defined above. We use two strategies in the construction: iterative processes similar to the ones in [24, 19], and double covers of the SG dependencies. These two strategies allow us to construct algebras generated by $\text{poly}(1/\delta)$ many elements with the desired properties for each of the subsets above. The iterative processes allow us to tightly control $\mathcal{F}_{\text{square}}$ and obtain some control over $\mathcal{F}_{\text{linear}}$ and $\mathcal{F}_{\text{span}}$, whereas the double covers allow us to handle \mathcal{F}_{deg} and also to prove that the remaining linear forms will become a δ -LCC configuration.

The final small subalgebra. Once we have our clean subalgebra with respect to \mathcal{F} , and every polynomial in \mathcal{F} either in the subalgebra or univariate over our subalgebra, we proceed to prove that the “additional linear forms” that arise in this way, together with the linear forms from our configuration, span a vector space of small dimension. While these linear forms satisfy linear relations, the linear forms corresponding to different quadratics in our set might be the same, and therefore the set of linear forms might not form a robust linear Sylvester-Gallai configuration. However, the fact that the vector space V is saturated implies that not too many quadratics have the same linear form: if they did, then we could add that linear form to V_1 and add many polynomials of \mathcal{F} to $\mathbb{C}[V]$. This saturation allows us to show that the linear forms form a δ -LCC configuration, and therefore span a vector space of small dimension. We extend our algebra $\mathbb{C}[V]$ by adjoining the generators of this small vector space to obtain the final small subalgebra containing \mathcal{F} as desired.

1.3 Related work

The original motivation for studying higher degree SG configurations comes from [11], in order to give polynomial time PIT for a special class of depth-4 algebraic circuits. The most general SG problem/configuration that is needed towards this application is the following conjecture. As we mentioned earlier, Conjecture 1 is a first step towards the proof of this conjecture. The most general form of Gupta’s conjecture [11, Conjecture 1], which we term as (k, d, c) -Sylvester-Gallai conjecture, is stated below.

► **Conjecture 5** ((k, d, c) -Sylvester-Gallai conjecture). *Let $k, d, c \in \mathbb{N}^*$ be parameters, and let $\mathcal{F}_1, \dots, \mathcal{F}_k$ be finite sets of irreducible polynomials of degree at most d such that*

- $\cap_i \mathcal{F}_i = \emptyset$,
- *for every Q_1, \dots, Q_{k-1} each from a distinct set \mathcal{F}_{i_j} , there are polynomials P_1, \dots, P_c in the remaining set such that $\prod P_i \in \text{rad}(Q_1, \dots, Q_{k-1})$.*

Then the transcendence degree of the union $\cup_i \mathcal{F}_i$ is a function of k, d, c , independent of the number of variables or the size of the sets \mathcal{F}_i .

As a step towards the proof of Conjecture 5, [24] studies quadratic Sylvester-Gallai configurations (Conjecture 1). The configurations we study are exactly the fractional versions of these quadratic Sylvester-Gallai configurations. In [19], the authors extend the result on quadratic Sylvester-Gallai configurations, weakening the Sylvester-Gallai condition, only requiring that the radical of the ideal generated by every pair of quadratics contains a product of four other quadratics. In [20], the authors extend this further, by proving Conjecture 5 for the case of $k = 3, d = 2$ and $c = 4$, which gives a polynomial time blackbox PIT for algebraic circuits computing a sum of three products of quadratic polynomials.

Our proof techniques and intermediate results generalise some of those of [24], [19], [20]. In [24], the author proves a structural result for quadratic forms contained in the radical of the ideal generated by two other quadratic forms. In [19], this result is extended to products

of quadratics. Our structure result directly classifies the radical of the ideal generated by two quadratics based on the number and degree of the minimal primes of the ideal. Both structure theorems of [24] and [19] follow as immediate corollaries.

Further, our definition of clean vector spaces and the clean up procedure is a generalisation of part of the strategy in the above works. In [24, 19], the authors construct two vector spaces: one of linear forms and another of quadratic forms, and then they prove that most polynomials in the configuration can be written as the sum of a quadratic polynomial in the second vector space, and a polynomial “close” to the algebra generated by the first vector space. Our definition of clean vector spaces formalizes this strategy, giving us more structure which helps us unify the case analysis in these previous works.

Another important point to notice is that in this paper we do not make use of the projection trick used in [24, 19]. While the parameters become slightly worse for not using the projection trick, as we now have to account for repetitions in the set of linear forms not in the algebra, we believe that getting rid of the projection trick will make this strategy more amenable to generalizations to higher degree.

Progress on Polynomial Identity Testing. Recently, there has been remarkable progress on the PIT problem for depth 4 circuits (the same algebraic circuits considered in [11]). In [6], the authors give a quasi-polynomial time algorithm for blackbox PIT for depth 4 circuits with bounded top and bottom fanins. Their approach involves considering the logarithmic derivative of circuits, and is analytic in nature, which allows them to bypass the need of Sylvester-Gallai configurations. Another PIT result in this setting comes from the lower bound against low depth algebraic circuits proved by [16], which gives a weakly-exponential algorithm for PIT for these circuits via the hardness vs randomness paradigm for constant depth circuits [3]. However, the SG approach of [11] is the only one so far which could yield polynomial-time blackbox PIT algorithms for the subclass of depth-4 circuits with constant top and bottom fanins.

Comparison with [18]. In [18] the authors prove the radical SG theorem for cubic forms (not the robust version). Their work is on one hand more general, since they are now handling cubic forms as well, but it is less general in that their SG theorem is not robust, and the robustness - as we have seen, significantly increases the complexity of the problem (as it is the case in every setting, even in the linear case). In their work, the authors proceed with a similar strategy as the previous works and this one, by proving a structure theorem for ideals generated by two cubics, and then constructing a “robust algebra” where the forms become “univariate” with respect to it. Some of the ideas in this paper are motivated by similar constructions done in their work. More precisely, their construction of wide algebras motivated our construction of clean vector spaces, where the difference between the constructions is that in their work they need stronger algebro-geometric properties of their algebras, but to achieve that their algebras must be significantly larger than the ones we construct in this paper. Apart from this motivation, both works are distinct in their techniques, since in our case the robustness severely constrains our choice of dependencies.

Simultaneous result [21]. Simultaneously and independently from this work, Peleg and Shpilka have also proved that $(\delta, 2)$ -rad-SG configurations have $\text{poly}(1/\delta)$ dimension. While the result of [21] in its current form works when the configuration only has irreducible quadratics, in our work we also allow linear forms in our configurations.

There are a number of parallels between the methods used in [21] and the ones used in our paper. Both use structure theorems for ideals generated by quadratics, and structure theorems for (x, y) -primary ideals. Further, both results divide the configuration into special sets based on the cases of the structure theorem, and control each of these sets separately.

One key technical difference between our approach and [21] is the structure used to control the above sets. In [21] they use an algebra generated by linear forms and quadratics with the property that linear combinations of quadratics are high rank even after taking quotients with the linear forms. We define the notion of clean vector spaces, which generate “special algebras” which apart from having the above property (what we call robustness) are also saturated in the sense that adding a few linear forms cannot bring too many polynomials in our configuration “closer” to the vector space. We also use the notion of univariate polynomials over clean vector spaces, and prove the existence of a small clean vector space V such that the polynomials in each special set is univariate over V . Once we have such structure, we can assign to each univariate polynomial a linear form ℓ_i (the “extra variable” from this polynomial), and we then show that the set of linear forms $\{\ell_i\}$ corresponding to each polynomial forms a LCC configuration.

Another key technical difference is that in our work, we **do not** make use of the *projection method*, as we believe that in higher degrees such method may not be amenable to generalization without generalizing the SG conjectures as well. This is one of the main reasons why we can only prove that the univariate polynomials ℓ_i form a LCC configuration, instead of a robust linear SG configuration. This in turn is also the reason that our bound is worse than the one in [21].

Handling the linear forms presents an extra technical challenge. The main difficulty arises when a quadratic Q satisfies the SG condition with many linear forms ℓ , as there is less structure between Q, ℓ and the quadratic in $\text{rad}(Q, \ell)$ than when the configurations just consist of quadratics. This lack of structure makes our analysis significantly more intricate.

1.4 Organization

In Section 2 we state the formal definitions of robust and clean vector spaces. In Section 3 we state the condition we want our small algebra (as described in Section 1.2) to satisfy, and how this implies the main theorem. Finally, in Section 4 we state some concluding remarks, and list a number of open problems and further directions. Due to space limitations, all proofs and detailed discussions are omitted from this article, and can be found in the full version on arxiv.

2 Clean vector spaces

In this section we formally define the notions of robust and clean vector spaces as described in the introduction. We refer to the full version for examples, further details and related statements. We begin with a definition of polynomials which are close to being in the algebra generated by a vector space of forms. Recall that $S = \mathbb{C}[x_1, \dots, x_n]$, and that S_d refers to the vector space of polynomials of degree d .

► **Definition 6** (Polynomials close to a vector space). *Given a vector space $V = V_1 + V_2$ where $V_i \subseteq S_i$ we say that a quadratic P is s -close to V if there is a polynomial $Q \in \mathbb{C}[V]$ such that $\text{rank}(P - Q) = s$, and for any polynomial $Q' \in \mathbb{C}[V]$, we have that $\text{rank}(P - Q') \geq s$. If a polynomial P is not r -close to V , for any $r \leq s$, we say that P is s -far from V .*

42:10 Robust Radical Sylvester-Gallai Theorem for Quadratics

With the definition above in hand, we are ready to define robust vector spaces. These are vector spaces whose quadratic forms are in a sense far from the ideal generated by the linear forms.

► **Definition 7** (Robust vector spaces). *A vector space $V = V_1 + V_2$ where $V_i \subseteq S_i$ is said to be r -robust if, for any nonzero $Q \in V_2$, the following conditions hold:*

1. Q is $(r - 1)$ -far from V_1
2. if $Q \notin (V_1)$, then $\text{rank}(\overline{Q}) \geq r$, where $\overline{Q} \in S/(V_1)$ denotes the image of Q in the quotient ring $S/(V_1)$.

If a homogeneous ideal I has a generating set $V_1 + V_2$ which is r -robust, we say that I is an r -robust ideal.

In the above definition, we use the fact that the quotient ring $S/(V_1)$ is isomorphic to a polynomial ring, in order to define $\text{rank}(\overline{Q})$. Next we define the relative vector space of a quadratic form and the notion of a polynomial being univariate over a vector space. We refer to the full version for statements regarding well-definedness of these notions.

► **Definition 8** (Vector space of a quadratic form). *Let Q be a quadratic form of rank s , so that $Q = \sum_{i=1}^s a_i b_i$. Define the vector space $\mathbb{L}(Q) := \text{span}_{\mathbb{C}} \{a_1, \dots, a_s, b_1, \dots, b_s\}$. Define $\mathbb{L}(Q)$ as:*

$$\mathbb{L}(Q) = \begin{cases} \text{span}_{\mathbb{C}} \{Q\}, & \text{if } s \geq 5 \\ \mathbb{L}(Q), & \text{otherwise.} \end{cases}$$

► **Definition 9** (Relative space of linear forms). *If V is an r -robust vector space and P is s -close to V for $s < r/2$ we can define*

$$\mathbb{L}_V(P) := \begin{cases} \mathbb{L}(P - Q) + V_1, & \text{if } s \leq 4 \\ \text{span}_{\mathbb{C}} \{P\}, & \text{otherwise} \end{cases}$$

where $Q \in \mathbb{C}[V]$ is a polynomial such that $\text{rank}(P - Q) = s$. We also define the quotient space

$$\overline{\mathbb{L}}_V(P) := \begin{cases} \mathbb{L}_V(P)/V_1, & \text{if } s \leq 4 \\ 0, & \text{otherwise} \end{cases}$$

► **Definition 10** (Univariate polynomials over robust vector spaces). *Let $V := V_1 + V_2$ be an r -robust vector space, where $r \geq 3$ and $V_i \subseteq S_i$, for $i \in \{1, 2\}$. We say that a form P is univariate over V if P is 1-close to V and $\dim(\overline{\mathbb{L}}_V(P)) = 1$. Moreover, we define $z_P \in S_1/V_1$ to be the linear form such that $\overline{\mathbb{L}}_V(P) := \text{span}_{\mathbb{C}} \{z_P\}$.*

We are now ready to define the main object of this section: *clean vector spaces*. The subalgebras that we construct in the proof of the Theorem 3 will be algebras generated by clean vector spaces. The cleanliness conditions imply that the quadratic generators are of high rank and that one can not add a small number of linear forms to a clean vector space V to increase the algebra $\mathbb{C}[V]$ such that the new algebra contains a lot of new polynomials from \mathcal{F} . The second condition will be the key to reduce the radical-SG-condition to a linear-SG condition once we show that our polynomials are univariate over a clean vector space.

► **Definition 11** (Clean vector spaces). Let $\mathcal{F} := \{Q_1, \dots, Q_m\} \subset S_1 \cup S_2$ be a set of forms and $r \geq 17$ be an integer. Let $V = V_1 + V_2$ be a vector space with $V_i \subset S_i$. We say that V is an (r, ε) -clean vector space over \mathcal{F} if the following conditions hold:

1. V is an r -robust vector space
2. For any $U_1 \subset S_1$ such that $\dim(U_1) \leq 8$, there are $< \varepsilon m$ polynomials $Q_j \in \mathcal{F}$ such that Q_j is s -close to V for $1 \leq s \leq 4$ and

$$\dim(\overline{\mathbb{L}}_V(Q_j)) > \dim(\overline{\mathbb{L}}_{V+U_1}(Q_j)).$$

If $V = V_1 + V_2$ is an (r, ε) -clean vector space over \mathcal{F} , then we say that the ideal (V) is an (r, ε) -clean ideal over \mathcal{F} , and similarly the algebra $\mathbb{C}[V]$ is an (r, ε) -clean algebra over \mathcal{F} .

3 Proof of Theorem 3

In this section we formalize the sketch of the proof given in the introduction. We refer to the full version for the formal definitions and proofs of the lemmas below. We use the partition of $\mathcal{F}_2 = \mathcal{F}_{\text{span}} \cup \mathcal{F}_{\text{linear}} \cup \mathcal{F}_{\text{square}} \cup \mathcal{F}_{\text{deg}}$ as defined in the introduction. The first step is to construct an intermediate small algebra such that any polynomial from our configuration $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ is either in the algebra, or univariate over this algebra (Lemma 12). The second step is to prove that we can augment this algebra slightly to contain all forms from \mathcal{F} (Lemma 13). We achieve this by showing that the extra variables corresponding to the polynomials form a LCC configuration, allowing us to bound their rank.

► **Lemma 12** (Reduction to Base Configuration). Let $0 < \delta \leq 1$ be a constant, and let $\varepsilon := \delta/10$. Let \mathcal{F} be a $(\delta, 2)$ -rad-SG configuration. There exists a $(17, \varepsilon^3/4^8)$ -clean vector space with respect to \mathcal{F} , denoted by V , such that every polynomial in \mathcal{F} is either in $\mathbb{C}[V]$ or univariate over V , and $\dim(V) = O(1/\varepsilon^4)$. Further, $\mathcal{F}_{\text{square}} \cup \mathcal{F}_{\text{linear}} \subseteq \mathbb{C}[V]$. Also, for every polynomial $P \in \mathcal{F}_{\text{deg}} \setminus \mathbb{C}[V]$, if z_P spans $\overline{\mathbb{L}}_V(P)$ then there are at least $\varepsilon^3 m/4^8$ distinct linear forms x_1, \dots, x_t and distinct linear forms a_1, \dots, a_t such that for every i , the linear forms z_P, x_i, a_i are pairwise linearly independent in S_1/V_1 , and $z_P \in \text{span}_{\mathbb{C}}\{x_i, a_i\}$.

► **Lemma 13** (Base Configuration). Let $0 < \delta \leq 1$, let $\varepsilon := \delta/10$ and let $0 < \gamma \leq \varepsilon^3/4^8$ be constants. If \mathcal{F} is a $(\delta, 2)$ -rad-SG configuration, and $V := V_1 + V_2$ is a $(17, \gamma)$ -clean vector space with respect to \mathcal{F} that satisfies the conditions of Lemma 12, then there exists $U \subset S_1$ with $\dim(U) = O(1/\varepsilon^{27})$ such that $\mathcal{F} \subset \mathbb{C}[V, U]$.

► **Theorem 3** ($(\delta, 2)$ -rad-SG theorem). If \mathcal{F} is a $(\delta, 2)$ -rad-SG configuration, then

$$\dim(\text{span}_{\mathbb{C}}\{\mathcal{F}\}) = O(1/\delta^{54}).$$

Proof. We use the previous two lemmas to prove the main theorem. Let $\varepsilon := \delta/10$, Given a $(\delta, 2)$ -rad-SG, we first apply Lemma 12 to obtain V , a $(17, \varepsilon^3/4^7)$ -clean vector space with respect to \mathcal{F} . The space V has dimension $\mathcal{O}(1/\varepsilon^4)$, and is such that every polynomial in \mathcal{F} is either in the algebra $\mathbb{C}[V]$, or univariate over V . We now apply Lemma 13 with parameter $\gamma = \varepsilon^3/4^8$ and vector space V , to obtain a vector space $U \subseteq S_1$. The vector space U has dimension $\mathcal{O}(1/\varepsilon^{27})$, and is such that $\mathcal{F} \subseteq \mathbb{C}[V, U]$.

Consider the algebra $\mathbb{C}[V, U]$. Since the generators are homogeneous, the set of linear forms $\mathbb{C}[V, U]_1$ in the vector space $U + V_1$. Further, every quadratic in this algebra is a linear combination of elements of V_2 , and products of the form $\ell_1 \ell_2$, where $\ell_i \in U + V_1$. Therefore, we have $\mathbb{C}[V, U]_2 = \mathcal{O}(1/\varepsilon^{54})$. The vector space $\mathbb{C}[V, U]_1 + \mathbb{C}[V, U]_2$ contains \mathcal{F} and has dimension $\mathcal{O}(1/\varepsilon^{54})$. This completes the proof. ◀

4 Conclusion and open problems

In this paper, we prove a robust version of the radical Sylvester-Gallai theorem for quadratics, generalizing [24].

Just as in the linear case of the Sylvester-Gallai problem robustness plays an important role in generalizing Sylvester-Gallai results to higher dimensional variants, such as the flats version in [1], we expect our robust variant to allow us to generalize the Sylvester-Gallai problem to “higher codimension” tuples of quadratic polynomials. For instance, instead of requiring $\text{rad}(F_i, F_j)$ to intersect \mathcal{F} non-trivially, one would only require that for many triples (i, j, k) , we would require $\text{rad}(F_i, F_j, F_k)$ to intersect \mathcal{F} non-trivially. Just as in the linear case, properly defining such higher codimension variants requires some careful thought, especially since the non-linear aspect will introduce more subtlety than the linear case.⁷ These higher dimensional variants have applications in algebraic complexity, as they can be instrumental in proving the main conjectures posed in [11] about such SG configurations.

Another important open problem is to generalize the above result to prove a robust version of the “product version” of the Sylvester-Gallai problem - a robust version of [11, Conjecture 1] with $k = 3$ and $r = 2$. In this work, we made a somewhat strong use of the fact that we have an extra polynomial in the radical ideal, and having a product of polynomials in the ideal instead seems to require a strengthening of several arguments in this paper to address it. Just as in [19], we believe that our general structure theorem, which gives us a deeper look in the minimal primes, could shed some light into a different way to construct robust algebras.

It is important to remark that higher codimension variants of the Sylvester-Gallai problem, even for quadratics, involves the study of schemes which are not equidimensional, which may require stronger structural results on the structure of such ideals. However, one could hope that our structure theorems might suffice, just as in [1] the robust linear Sylvester-Gallai theorem was sufficient to induct on the higher-dimensional analogs.

Lastly, another interesting direction and potential application of robust SG configurations is in the study of non-linear locally correctable codes (LCCs) over fields of characteristic zero. While lower bounds for linear LCCs have been out of reach for current techniques even over characteristic zero,⁸ it would be interesting to know if robust non-linear SG configurations have bounded transcendence degree. If a robust form of Gupta’s general conjecture is false, it could yield the first constructions of non-linear LCCs over characteristic zero, which are not known to exist. Moreover, we currently do not know of any construction of such codes with constant queries over characteristic zero.

References

- 1 Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC ’11, pages 519–528, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1993636.1993705.
- 2 Peter Borwein and William OJ Moser. A survey of sylvester’s problem and its generalizations. *Aequationes Mathematicae*, 40(1):111–135, 1990.
- 3 Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure results for polynomial factorization. *Theory of Computing*, 15(1):1–34, 2019.

⁷ In [1] had to account for sub-flats intersecting \mathcal{F} non-trivially.



⁸ Aside from 2-query LCCs where optimal lower bounds are known for both linear and non-linear codes.

- 4 J-L Colliot-Thélène, J-J Sansuc, and P Swinnerton-Dyer. Intersections of two quadrics and châtelet surfaces. i. *Journal für die reine und angewandte Mathematik*, 373:37–107, 1987.
- 5 Leonard Eugene Dickson. The points of inflexion of a plane cubic curve. *The Annals of Mathematics*, 16(1/4):50–66, 1914.
- 6 Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. Deterministic identity testing paradigms for bounded top-fan-in depth-4 circuits. In *Proceedings of the 36th Computational Complexity Conference, CCC '21*, Dagstuhl, DEU, 2021. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CCC.2021.11.
- 7 Zeev Dvir. Incidence theorems and their applications. *arXiv preprint*, 2012. arXiv:1208.5073.
- 8 Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Improved rank bounds for design matrices and a new proof of kelly’s theorem. In *Forum of Mathematics, Sigma*, volume 2. Cambridge University Press, 2014.
- 9 Paul Erdos, Richard Bellman, Hubert S Wall, James Singer, and Victor Thébault. Problems for solution: 4065-4069. *The American Mathematical Monthly*, 50(1):65–66, 1943.
- 10 Tibor Gallai. Solution of problem 4065. *American Mathematical Monthly*, 51:169–171, 1944.
- 11 Ankit Gupta. Algebraic geometric techniques for depth-4 pit & sylvester-gallai conjectures for varieties. In *Electron. Colloquium Comput. Complex.*, volume 21, page 130, 2014.
- 12 Sten Hansen. A generalization of a theorem of sylvester on the lines determined by a finite point set. *Mathematica Scandinavica*, 16(2):175–180, 1965.
- 13 William Vallance Douglas Hodge and Daniel Pedoe. *Methods of Algebraic Geometry: Volume 2*. Cambridge University Press, 1994.
- 14 Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 198–207. IEEE, 2009.
- 15 Leroy Milton Kelly. A resolution of the sylvester-gallai problem of j.-p. serre. *Discrete & Computational Geometry*, 1(2):101–104, 1986.
- 16 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. *Electron. Colloquium Comput. Complex.*, page 81, 2021. URL: <https://eccc.weizmann.ac.il/report/2021/081>.
- 17 Eberhard Melchior. Über vielseitigkeit der projektiven ebene. *Deutsche Math*, 5:461–475, 1940.
- 18 Rafael Oliveira and Akash Sengupta. Radical sylvester-gallai theorem for cubics. *Manuscript*, 2021.
- 19 Shir Peleg and Amir Shpilka. A generalized sylvester-gallai type theorem for quadratic polynomials. *CoRR*, abs/2003.05152, 2020. arXiv:2003.05152.
- 20 Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits via edelstein-kelly type theorem for quadratic polynomials. *CoRR*, abs/2006.08263, 2020. arXiv:2006.08263.
- 21 Shir Peleg and Amir Shpilka. Robust sylvester-gallai type theorem for quadratic polynomials. *CoRR*, abs/2202.04932, 2022. arXiv:2202.04932.
- 22 Nitin Saxena and Comandur Seshadhri. From sylvester-gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *Journal of the ACM (JACM)*, 60(5):1–33, 2013.
- 23 Jean-Pierre Serre. Advanced problem 5359. *Amer. Math. Monthly*, 73(1):89, 1966.
- 24 Amir Shpilka. Sylvester-gallai type theorems for quadratic polynomials. *Discrete Analysis*, page 14492, 2020.
- 25 Gaurav Sinha. Reconstruction of real depth-3 circuits with top fan-in 2. In *31st Conference on Computational Complexity (CCC 2016)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.
- 26 James Joseph Sylvester. Mathematical question 11851. *Educational Times*, 59(98):256, 1893.
- 27 Endre Szemerédi and William T. Trotter. Extremal problems in discrete geometry. *Combinatorica*, 3(3):381–392, 1983.

Robust Sylvester-Gallai Type Theorem for Quadratic Polynomials

Shir Peleg  

Tel Aviv University, Israel

Amir Shpilka  

Tel Aviv University, Israel

Abstract

In this work we extend the robust version of the Sylvester-Gallai theorem, obtained by Barak, Dvir, Wigderson and Yehudayoff, and by Dvir, Saraf and Wigderson, to the case of quadratic polynomials. Specifically, we prove that if $\mathcal{Q} \subset \mathbb{C}[x_1, \dots, x_n]$ is a finite set, $|\mathcal{Q}| = m$, of irreducible quadratic polynomials that satisfy the following condition

There is $\delta > 0$ such that for every $Q \in \mathcal{Q}$ there are at least δm polynomials $P \in \mathcal{Q}$ such that whenever Q and P vanish then so does a third polynomial in $\mathcal{Q} \setminus \{Q, P\}$.

then $\dim(\text{span}\{\mathcal{Q}\}) = \text{Poly}(1/\delta)$.

The work of Barak et al. and Dvir et al. studied the case of linear polynomials and proved an upper bound of $O(1/\delta)$ on the dimension (in the first work an upper bound of $O(1/\delta^2)$ was given, which was improved to $O(1/\delta)$ in the second work).

2012 ACM Subject Classification Mathematics of computing \rightarrow Mathematical analysis; Theory of computation \rightarrow Computational geometry

Keywords and phrases Sylvester-Gallai theorem, quadratic polynomials, Algebraic computation

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.43

Related Version *Full Version:* <http://arxiv.org/abs/2202.04932>

Funding *Shir Peleg:* The research leading to these results has received funding from the Israel Science Foundation (grant number 514/20) and from the Len Blavatnik and the Blavatnik Family foundation.

Amir Shpilka: The research leading to these results has received funding from the Israel Science Foundation (grant number 514/20) and from the Len Blavatnik and the Blavatnik Family foundation.

Independent result

Independently of our work, [18] have also proved the same result. Both works have been presented in a common talk at CG week 2022. For a more detailed comparison between the works, we refer the reader to Subsection 1.2.

1 Introduction

In this paper we prove a robust version of a result of [40]: Let $\mathcal{T} \subset \mathbb{C}[x_1, \dots, x_n]$ be a finite set of polynomials. We say that $Q_1(\vec{x}), Q_2(\vec{x}) \in \mathcal{Q}$ satisfy the *Polynomial Sylvester-Gallai condition* (PSG-condition for short) if there is a third polynomial $Q_3(\vec{x}) \in \mathcal{Q}$ such that $Q_3(\vec{x})$ vanishes whenever $Q_1(\vec{x})$ and $Q_2(\vec{x})$ vanish. We prove that if $\mathcal{T} \subset \mathbb{C}[x_1, \dots, x_n]$ is a finite set containing only irreducible quadratic polynomials, such that for every $Q \in \mathcal{T}$ a δ fraction of the polynomials in \mathcal{T} satisfy the PSG-condition with Q , then $\dim(\text{span}\{\mathcal{T}\}) = \text{Poly}(1/\delta)$.



© Shir Peleg and Amir Shpilka;

licensed under Creative Commons License CC-BY 4.0

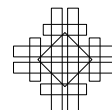
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 43; pp. 43:1–43:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



The motivation for proving this result, besides its own appeal, is two fold: a similar theorem played an important role in the polynomial identity testing (PIT for short) problem for small depth algebraic circuits, one of the fundamental open problems in theoretical computer science, see [33]; and it is also related to a long line of work extending and generalizing the original Sylvester-Gallai theorem [30, 17]. In particular, our result builds and generalizes a result of [3, 9], that can be viewed as proving an analogous claim for the case of degree-1 polynomials. Such results are useful in discrete geometry [3, 9], in the study of locally correctable codes, for reconstruction of certain depth-3 circuits [39, 25, 42] and more. See the survey of Dvir on incidence geometry for some applications of Sylvester-Gallai type theorems [7].

We next give background on the Sylvester-Gallai theorem, and some of its variants, and then discuss the connection to the polynomial identity testing problem.

Sylvester-Gallai type theorems

The Sylvester-Gallai theorem (SG-theorem) asserts that given a set $S = \{\vec{v}_1, \dots, \vec{v}_m\} \subset \mathbb{R}^n$ such that S is not contained in a line, there must be a line that contains exactly two points from S . It was first conjectured by Sylvester in 1893 [44] and then proved, independently, by Melchior in 1941 [30] and Gallai [17] in 1943 (in an answer to the same question posed by Erdős, who was unaware of Melchior’s result [12]). There are many extensions and generalizations of the theorem. We shall state a few that are related to this work. It is also helpful to think of the contra-positive statement. We say that a set of points is a Sylvester-Gallai configuration (SG-configuration for short) if every line that intersects the set at two points, must contain at least three points from the set. Thus, an SG-configuration in \mathbb{R}^n must be colinear.

In [38] Serre, aware that the original formulation of the theorem is not true over \mathbb{C} asked “Is there a nonplanar version of the Sylvester-Gallai configuration over the field of complex numbers?” Kelly proved that the answer is no, i.e. that every finite set of points in \mathbb{C}^n satisfying the SG-condition is planar [27]. Edelstein and Kelly proved a colorful variant of the problem: if three finite sets of points in \mathbb{R}^n satisfy that every line passing through points from two different sets also contains a point from the third set, then, the points belong to a three-dimensional affine space. This result can be extended to any constant number of sets. Many more extensions and generalizations of the SG-theorem are known, e.g. [22, 8]. The survey by Borwein and Moser [5] is a good resource on the SG-Theorem and some of the different variants that have been studied in the past.

More recently, Barak et al. [3] and Dvir, Saraf and Wigderson [9], motivated by questions on locally decodable codes and construction of rigid matrices, proved a *robust* (or fractional) version of the SG-theorem:

► **Definition 1** (δ -SG configuration). *We say that a set of points $v_1, \dots, v_m \in \mathbb{C}^n$ is a δ -SG configuration if for every $i \in [m]$ there exists at least $\delta(m-1)$ values of $j \in [m]$ such that the line through v_i, v_j contains a third point in the set.*

► **Theorem 2** (Theorem 1.9 of [9]). *Let $V = \{v_1, \dots, v_m\} \subset \mathbb{C}^n$ be a δ -SG configuration. Then $\dim(\text{span}\{v_1, \dots, v_m\}) \leq \frac{12}{\delta} + 1$.*

Algebraic generalizations of Sylvester-Gallai type theorems

Although the Sylvester-Gallai theorem and Theorem 2 are formulated in the setting of discrete geometry, there is a very natural algebraic formulation: If a finite set of pairwise linearly independent vectors, $\mathcal{S} \subset \mathbb{C}^n$, has the property that every two vectors span a third vector in the set, then the dimension of \mathcal{S} is at most 3. The proof is immediate from Kelly's theorem: pick a subspace H of codimension 1, which is in general position with respect to the vectors in \mathcal{S} . The intersection points $p_i = H \cap \text{span}\{s_i\}$, for $s_i \in \mathcal{S}$, satisfy the SG-condition over \mathbb{C} . Therefore, $\dim(\mathcal{S}) \leq 3$. An equivalent formulation, in the case of linear functions, is the following: If a finite set of pairwise linearly independent linear forms, $\mathcal{L} \subset \mathbb{C}[x_1, \dots, x_n]$, has the property that for every two forms $\ell_i, \ell_j \in \mathcal{L}$ there is a third $\ell_k \in \mathcal{L}$, such that $\ell_k = 0$ whenever $\ell_i = \ell_j = 0$, then the linear dimension of \mathcal{L} is at most 3. To see the equivalence note that it must be the case that $\ell_k \in \text{span}\{\ell_i, \ell_j\}$ and thus the coefficient vectors of the forms in the set satisfy the condition for the (vector version of the) SG-theorem, and the bound on the dimension follows. Observe that the last example shows that in the case of linear functions the PSG-condition and the SG-condition are equivalent. The last formulation can now be generalized to higher degree polynomials. In particular, the following conjecture was raised by Gupta [20].

► **Definition 3** (PSG-configuration). *Let $\mathcal{T} \subset \mathbb{C}[x_1, \dots, x_n]$ be a set of polynomials. We say that $Q_1, Q_2 \in \mathcal{T}$ satisfy the Polynomial Sylvester-Gallai condition (PSG-condition for short) if there is a third polynomial $Q_3(\vec{x}) \in \mathcal{T}$ such that Q_3 vanishes whenever Q_1 and Q_2 vanish.*

We say that a set \mathcal{T} is a PSG-configuration if every two polynomials $Q_1, Q_2 \in \mathcal{T}$ satisfy the PSG-condition.

► **Problem 1** (Conjecture 2 of [20]). *There is a function $\lambda : \mathbb{N} \rightarrow \mathbb{N}$ such that for any finite set $\mathcal{T} \subset \mathbb{C}[x_1, \dots, x_n]$ of pairwise linearly independent and irreducible polynomials, of degree at most r , that satisfy the PSG-condition, it holds that the algebraic rank of \mathcal{T} is at most $\lambda(r)$.*

This problem was answered affirmatively, with a stronger conclusion, in the case of quadratic polynomials ($r = 2$) in [40].

► **Theorem 4** (Theorem 1.7 of [40]). *There is a constant λ such that the following holds for every $n \in \mathbb{N}$. Let $\mathcal{T} \subset \mathbb{C}[x_1, \dots, x_n]$ consist of homogeneous quadratic polynomials, such that each $Q \in \mathcal{T}$ is either irreducible or a square of a linear function. If \mathcal{T} satisfies the PSG-condition then $\dim(\text{span}\{\mathcal{T}\}) \leq \lambda$.*

Motivated by applications for the polynomial identity testing problem, Gupta [20] and Beecken, Mittmann and Saxena [4] also raised the following colorful variant, which generalizes the Edelstein-Kelly theorem.

► **Conjecture 5** (Conjecture 30 of [20]). *There is a function $\lambda : \mathbb{N} \rightarrow \mathbb{N}$ such that the following holds for every $r, n \in \mathbb{N}$. Let R, B, G be finite disjoint sets of pairwise linearly independent, irreducible, homogeneous polynomials in $\mathbb{C}[x_1, \dots, x_n]$ of degree $\leq r$ such that for every pair Q_1, Q_2 from distinct sets there is a Q_3 in the remaining set so that whenever Q_1 and Q_2 vanish then also Q_3 vanishes. Then the algebraic rank of $(R \cup B \cup G)$ is at most $\lambda(r)$.*

This problem was also answered affirmatively, with the same stronger conclusion, in [40], for the case of quadratic polynomials.

► **Theorem 6** (Theorem 1.8 of [40]). *There is a constant λ such that the following holds for every $n \in \mathbb{N}$. Let $\mathcal{T}_1, \mathcal{T}_2$ and \mathcal{T}_3 be finite sets of homogeneous quadratic polynomials over \mathbb{C} satisfying the following properties:*

- *Each $Q \in \cup_i \mathcal{T}_i$ is either irreducible or a square of a linear function.*
- *No two polynomials are multiples of each other (i.e., every pair is linearly independent).*
- *For every two polynomials Q_1 and Q_2 from distinct sets there is a polynomial Q_3 in the third set so that whenever Q_1 and Q_2 vanish then also Q_3 vanishes.*

Then $\dim(\text{span}\{\cup_i \mathcal{T}_i\})$ has dimension $O(1)$.

PIT and Sylvester-Gallai type theorems

The PIT problem asks to give a deterministic algorithm that given an arithmetic circuit as input determines whether it computes the identically zero polynomial. The circuit can be given either via a description of its graph of computation (white-box model) or via oracle access to the polynomial that it computes (black-box model). This is a fundamental problem in theoretical computer science that has received a lot of attention from researchers in the last two decades. Besides of being a natural and elegant question, the PIT problem is important due its connections to lower bounds for arithmetic circuits (hardness-randomness tradeoffs) [23, 1, 24, 11, 6]; its relation to other derandomization problems such as finding perfect matching deterministically, in parallel, [13, 43], derandomizing factoring algorithms [28], derandomization questions in geometric complexity theory [31, 15]; its role in algebraic natural proofs [16, 19]. In particular, PIT appears to be the most general algebraic derandomization problem. For more on the PIT problem see [41, 34, 35, 14]. For a survey of algebraic hardness-randomness tradeoffs see [29].

A beautiful line of work has shown that deterministic algorithms for the PIT problem for homogeneous depth-4 circuits or for depth-3 circuits would lead to deterministic algorithms for general circuits [2, 21]. This makes small depth circuits extremely interesting for the PIT problem. This is also the setting where Sylvester-Gallai type theorems play an important role. The relation between (colored-versions of the) SG-theorem and deterministic PIT algorithms for depth-3 circuits was observed in [10]. The work of [26, 37] used this relation to obtain polynomial- and quasi-polynomial-time PIT algorithms for depth-3 circuits, depending on the characteristic. Currently, the best algorithm for PIT of depth-3 circuits was obtained through a different yet highly related approach in [36]. As the SG-theorem played such an important role in derandomizing PIT for depth-3 circuits, it was asked whether a similar approach could work for depth-4 circuits. This motivated [4, 20] to raise Problem 1 and Conjecture 5. In [33] we gave a positive answer to Conjecture 5 for the case of degree-2 polynomials ($r = 2$). Interestingly, Theorem 2 played a crucial role in the proof, as well as in the proofs of [40, 32]. Studying the proofs of [40, 32, 33] leads to the conclusion that in order to solve Problem 1 and Conjecture 5 for degrees larger than 2, we must first obtain a result analogous to Theorem 2.

Our results

In this work we prove an analog of Theorem 2 for quadratic polynomials. We hope that this result will lead to an extension of the works [40, 32, 33] to higher degree polynomials.

► **Definition 7** (δ -PSG-configuration). *Let $\mathcal{Q} \subset \mathbb{C}[x_1, \dots, x_n]$ be a set of polynomials. We say that a finite set of polynomials \mathcal{Q} is a δ -PSG configuration if for every $Q \in \mathcal{Q}$ there are at least $\delta \cdot |\mathcal{Q}|$ polynomials $P \in \mathcal{Q}$ such that Q and P satisfy the PSG condition.*

► **Theorem 8.** *Let $\mathcal{Q} \subset \mathbb{C}[x_1, \dots, x_n]$ a finite set of irreducible quadratic polynomials. If \mathcal{Q} is a δ -PSG configuration then $\dim(\text{span}\{\mathcal{Q}\}) = O(1/\delta^{16})$.*

► **Remark 9.** The same conclusion holds even if we allow irreducible polynomials of degree at most 2 (i.e. if we allow linear functions). The proof is similar in nature, with more case analysis, and so we decided to omit it.

Note that this is robust version of Theorem 4 in the same sense that Theorem 2 is a robust version of the SG-theorem.

► **Remark 10.** While the result in Theorem 2 tight (up to the constant in the big Oh), we do not believe that the result of Theorem 8 is tight. In particular, we believe that the upper bound should be $O(1/\delta)$.

1.1 Proof idea

To explain the proof we will use some algebraic notations, $\langle \cdot \rangle$ denotes an ideal, $\sqrt{\langle \cdot \rangle}$ denotes the radical of the ideal, and $\mathbb{C}[V]_2$ denotes the space of all quadratic polynomials defined only using the linear forms in V .

At the heart of all previous work lies an algebraic theorem, classifying the cases in which a quadratic polynomial vanishes when two other quadratics vanish (actually, for [32, 33] a more general result was needed - a characterization of the different cases in which a product of quadratic polynomials vanishes whenever two other quadratics vanish).

► **Theorem 11** (Theorem 1.10 of [40]). *Let A, B and C be n -variate, homogeneous, quadratic polynomials, over \mathbb{C} , such that whenever A and B vanish then so does C . Then, one of the following cases must hold:*

- (i) *C is in the linear span of A and B .*
- (ii) *There exists a non trivial linear combination of the form $\alpha A + \beta B = \ell^2$ for some linear form ℓ .*
- (iii) *There exist two linear forms ℓ_1 and ℓ_2 such that when setting $\ell_1 = \ell_2 = 0$ we get that A and B (and consequently C) vanish.*

The high level idea in the proof of Theorem 4 (which was generalized in [32, 33]), includes two steps; The first step constructs a linear space of linear forms V , and a subset $\mathcal{J} \subset \mathcal{Q}$, both of constant dimension such that a vast majority of the polynomials in \mathcal{Q} are in $\text{span}\{\mathcal{J}, \mathcal{Q} \cap \langle V \rangle\}$.¹ Implementing this idea requires a lot of case analysis, according to Theorem 11. In the second step the dimension of $\mathcal{Q} \cap \langle V \rangle$ is upper bounded.

The idea outlined above heavily relies on the fact that when $\delta = 1$, the set $\mathcal{Q} \cap \langle V \rangle$ is a PSG-configuration in itself. Indeed, let $Q_1, Q_2 \in \mathcal{Q} \cap \langle V \rangle$. When $\delta = 1$ it follows that there is $Q_3 \in \mathcal{Q}$ such that $Q_3 \in \sqrt{\langle Q_1, Q_2 \rangle} \subseteq \langle V \rangle$. In order to bound the dimension of $\mathcal{Q} \cap \langle V \rangle$, [40] “projected” V to a one dimensional space $\text{span}\{z\}$ (where z is a new variable). Every polynomial $Q_i \in \mathcal{Q} \cap \langle V \rangle$ is mapped to a polynomial of the form $z \cdot \ell_i$, for some linear form ℓ_i . Then, it is proved that the ℓ_i ’s form an SG-condition.²

This technique fails when $\delta \in (0, 1)$. First, we cannot expect to prove that $\mathcal{Q} \cap \langle V \rangle$ is a δ' -PSG configuration by itself (even when we allow smaller, yet fixed, $\delta' \leq \delta$). For example, since $\delta < 1$, it may be the case that (many polynomials) $Q \in \mathcal{Q} \cap \langle V \rangle$ have *all* of

¹ [40] had different notations, and $|\mathcal{J}| = 1$.

² The reader should take note that this is a very high-level simplification of one part in the proof. For more details see the “easy-case” in [32, 33].

their neighbors outside $\mathcal{Q} \cap \langle V \rangle$. Furthermore, even if we knew that $\mathcal{Q} \cap \langle V \rangle$ is a δ' -PSG configuration, then it is not clear that by following the lines of [40] and mapping $\langle V \rangle$ to $\text{span}\{z\}$, the resulting ℓ 's, form a δ' -PSG configurations. The reason for that is a bit subtle: note that it may be the case that many polynomials $Q \in \mathcal{Q} \cap \langle V \rangle$ were mapped to $\text{span}\{z^2\}$. Thus, it may be the case that all the neighbors of some $z \cdot \ell$ are in $\text{span}\{z^2\}$, which gives us no information at all about ℓ . In contrast, in [40], since $\delta = 1$, we could get information about ℓ by its interaction with polynomials not in $\text{span}\{z^2\}$.

In order to overcome these issues, we needed to develop new techniques, and improve the characterization given in Theorem 11iii (see Corollary 19). Next, we present the outline of the proof in more details.

We start with the same line of constructing a linear space of linear forms V , and a subset $\mathcal{J} \subset \mathcal{Q}$, both of dimension $O(\text{poly}(\frac{1}{\delta}))$ such that $\mathcal{Q} \subseteq \text{span}\{\mathcal{J}, \langle V \rangle\}$. We partition \mathcal{Q} to four sets: $\mathcal{C}_{[V]} = \mathcal{Q} \cap \mathbb{C}[V]_2$; $\mathcal{C}_{\langle V \rangle} = (\mathcal{Q} \cap \langle V \rangle) \setminus \mathcal{C}_{[V]}$; $\mathcal{J}_{[V]} = \mathcal{Q} \cap \text{span}\{\mathcal{J} \cup \mathbb{C}[V]_2\}$; and the remaining set $\mathcal{J}_{\langle V \rangle} = \mathcal{Q} \cap \text{span}\{\mathcal{J} \cup \langle V \rangle\} \setminus \mathcal{J}_{[V]}$. We already know that $\dim(\mathcal{C}_{[V]} \cup \mathcal{J}_{[V]})$ is small, so we only have to bound the dimension of $\mathcal{C}_{\langle V \rangle} \cup \mathcal{J}_{\langle V \rangle}$.

Let us focus on $\mathcal{C}_{\langle V \rangle}$. We would like to prove that we can add a few linear functions to V to get a subspace U such that $\mathcal{C}_{\langle V \rangle} \subset \mathbb{C}[U]_2$. Let $P \in \mathcal{C}_{\langle V \rangle}$. First we consider the case that many of P 's neighbors (i.e. those polynomials with which P satisfies the PSG-condition) are in $\mathcal{C}_{[V]} \cup \mathcal{C}_{\langle V \rangle}$. To handle this case we strengthen Theorem 11iii and use it to show that if $Q \in \mathcal{C}_{[V]}$ is a neighbor of P then the polynomial $Q' \in \sqrt{\langle P, Q \rangle}$ is unique (see Corollary 18). This means that by moving the linear functions on which P depends to U , we move many polynomials from $\mathcal{C}_{\langle V \rangle}$ to $\mathbb{C}[V + U]_2$.

Next we consider the case where P has “many” neighbors in $\mathcal{J}_{[V]} \cup \mathcal{J}_{\langle V \rangle}$. To handle this case we first prove that P can only satisfy Theorem 11i with polynomials in $\mathcal{J}_{[V]} \cup \mathcal{J}_{\langle V \rangle}$. We prove that under this condition, there is a “large” subset of $\mathcal{C}_{\langle V \rangle}$ that is of constant dimension. Thus, by adding a few linear functions to U , we move many polynomials from $\mathcal{C}_{\langle V \rangle}$ to $\mathbb{C}[V + U]_2$ (see Claim 29). We can continue this process as long as $\mathcal{C}_{\langle V \rangle}$ is large enough, as the amount of polynomials that we move at any step depends on $|\mathcal{C}_{\langle V \rangle}|$. Therefore, when this process terminates we still have to deal with a set $\mathcal{C}_{\langle V \rangle}$ that is not large but not too small either (it is of size $\Omega(\delta m)$). Now, we turn our attention to $\mathcal{J}_{\langle V \rangle}$. Using similar arguments, and relying on the fact that $|\mathcal{C}_{\langle V \rangle}|$ is small, we prove that we can add a few linear functions to U and make $|\mathcal{J}_{\langle V+U \rangle}|$ small. Having achieved that, we prove that if both $|\mathcal{C}_{\langle V+U \rangle}|$ and $|\mathcal{J}_{\langle V+U \rangle}|$ are small then they are in fact, empty (see Claim 28).

1.2 The work of [18]

Independently from this work, Garg, Oliveira and Sengupta have also proved that δ -PSG configurations have dimension bounded by $\text{Poly}(\frac{1}{\delta})$.

While our result, in its current form, holds when the configuration is assumed to contain only irreducible quadratics, [18] also allow linear forms in the configuration. Our techniques are good enough to deduce the more general case, but since it adds more technical details that do not give more insight into the problem, we decided to omit that part of the proof.

There are a number of parallels between the methods used in [18] and the ones used in our paper. Both proofs use structure theorems that analyze the situation in which there is a quadratic polynomial in the radical of an ideal generated by two other quadratics. Basically those theorems prove that the involved quadratics must satisfy certain structural conditions. Further, both results partition the δ -PSG configuration to “special” sets based on the different cases of the structure theorem, and analyze each of these sets separately.

One key technical difference between our approach and that of [18] is the definition of these special sets. While we construct \mathcal{J}, V using our iterative process, [18] define the notion of clean vector spaces, which generate “special algebras” (in their terminology) that have similar properties, but are also saturated in the sense that adding a few linear forms to the vector space cannot bring too many polynomials from the configuration “closer” to the vector space. This is the analog of moving many polynomials from $\mathcal{J}_{(V)} \cup \mathcal{C}_{(V)}$ to $\mathcal{J}_{[V]} \cup \mathcal{C}_{[V]}$, until this cannot be done anymore, in our work.

[18] also uses the notion of univariate polynomials over clean vector spaces, whereas we work with the ideal generated by the vector space V . They show that there is a clean vector space W , of dimension at most $\text{Poly}(\frac{1}{\delta})$, such that the polynomials in each special set are univariate over W . In other words, each Q_i in the configuration can be represented as a polynomial in the space $\mathbb{C}[W, \ell_i]_2$ for some linear form ℓ_i . They then show that these ℓ_i 's form a LCC configuration (see [3, 18] for definition), instead of a robust linear SG configuration, which is what we use in our work. This in turn is also the reason why the bound in [18] is slightly worse than the one in our work.

1.3 Discussion

There are two distinct goal to the line of work [40, 32, 33], including this paper. The first is obtaining higher degree geometric extensions of the Sylvester-Gallai and Edelman-Kelley theorems. From the complexity theoretic point of view, the goal is to eventually obtain PIT algorithms for $\Sigma^{[k]}\Pi^{[d]}\Sigma\Pi^{[r]}$ circuits, for any $k, r = O(1)$. Currently we have a polynomial time PIT algorithm only for the case $k = 3$ and $r = 2$ [33]. To understand such a difficult question one has to start somewhere, and the case $k = 3$ and $r = 2$ was a natural starting point for the investigation (especially as no subexponential time PIT algorithm, even for $\Sigma^{[3]}\Pi^{[d]}\Sigma\Pi^{[2]}$ circuits, was known prior to [33]). Since so little is known, we believe that a natural approach for advancing is to first extend the results of [33] to higher degrees (i.e. higher values of r), and then for a higher top fan-in (i.e. higher values of k). Before we explain the difficulties in going to higher degrees we recall that [33] needed the following strengthening of Theorem 6 for their PIT algorithm.

► **Theorem 12** (Theorem 1.6 in [33]). *There exists a universal constant λ such that the following holds. Let $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 \subset \mathbb{C}[x_1, \dots, x_n]$ be finite sets of pairwise linearly independent homogeneous polynomials satisfying the following properties:*

- *Each $Q \in \cup_{j \in [3]} \mathcal{T}_j$ is either irreducible quadratic or a square of a linear function.*
- *Every two polynomials Q_1 and Q_2 from distinct sets satisfy that whenever they vanish then the product of all the polynomials in the third set vanishes as well.*

Then, $\dim(\text{span}\{\cup_{j \in [3]} \mathcal{T}_j\}) \leq \lambda$.

There are several difficult hurdles in going from $r = 2$ to general r , or even to $r = 3$, if we wish to continue working in the framework of [40, 32, 33] (and this paper). The first is understanding what is the correct generalization of Theorem 11 to higher degrees, as this theorem lies at the heart of all these papers. A second hurdle is obtaining a robust version of Theorem 12. First for $r = 2$ and then for higher degrees.

For extending Theorem 11 to higher degrees it seems natural to find an extension to $r = 3$. While it seems that such an approach could last forever and lead nowhere (as we will then have to prove a result for $r = 4$ etc.), we believe that understanding the case $r = 3$ can shed more light on the general case, as sometimes going from degree 2 to 3 is as difficult as the general case.

Once we prove such a structural theorem, we will need to extend Theorem 12 to higher values of r . An important tool in the proof of Theorem 12 was a robust version of the EK-theorem.

► **Definition 13** (δ -EK configuration). *We say that the sets $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 \subset \mathbb{C}^n$ form a δ -EK configuration if for every $i \in [3]$ and $p \in \mathcal{T}_i$ a δ fraction of the vectors q in the union of the two other sets satisfy that p and q span some vector in the third set (the one not containing p and q). We refer to a 1-EK configuration as simply an EK-configuration.*

► **Theorem 14** (Theorem 3.9 of [33]). *Let $0 < \delta \leq 1$ be any constant. Let $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 \subset \mathbb{C}^n$ be disjoint finite sets that form a δ -EK configuration. Then, $\dim(\text{span}\{\cup_i \mathcal{T}_i\}) = O(1/\delta^3)$.*

Thus, a natural continuation would be to prove a robust version of Theorem 14 for quadratic polynomials (i.e. a robust version of Theorem 6) and then to extend it to a robust version of Theorem 12 and to higher degrees. While in this paper we only prove a robust version of Theorem 4, we believe that with some more technical work this can be extended to a robust version of Theorem 6 as well. Hence, the next immediate challenge would be to obtain a robust version of Theorem 12 (or even of the main result of [32]). If we obtain such an extension and, in addition extend Theorem 11 to higher values of r , then we expect that a PIT algorithm for the case $k = 3$ and $r = 3$ would follow. More importantly, we believe that this will let us gain important understanding on how to generalize the results for arbitrary values of r .

2 Robust-SG theorems in \mathbb{C}^n

We shall need the following generalizations of Theorem 2. The proofs of this section appear in the full version.

► **Theorem 15.** *Let $0 < \delta \leq 1$ be any constant. Let $W \subset \mathbb{C}^n$ be an r -dimensional space. Let $\mathcal{W} \subset W$ and $\mathcal{K} \subset \mathbb{C}^n \setminus W$ be finite subsets such that no two vectors in $\mathcal{T} = \mathcal{K} \cup \mathcal{W}$ are linearly dependent. Assume further that all the elements in \mathcal{K} satisfy the following relaxed EK-property: For every $p \in \mathcal{K}$, for at least δ fraction of the points $q \in \mathcal{T}$ the span of p and q contains a point in $\mathcal{T} \setminus \{p, q\}$. Then, $\dim(\text{span}\{\mathcal{T}\}) \leq O(r + \frac{1}{\delta})$.*

We also use the following bi-partitive version of [9, Corollary 1.11] this is a slight variation of the formulation presented in their paper.

▷ **Claim 16.** Let $V = v_1, \dots, v_n \subset \mathbb{C}^d$ be a set of n distinct points. Suppose that there is $\mathcal{B} \subseteq V$ such that there are at least δn^2 pairs in $\mathcal{B} \times (V \setminus \mathcal{B})$ that lie on a special line. Then there exists a subset $\mathcal{B}' \subseteq \mathcal{B}$ such that $|\mathcal{B}'| \geq (\delta/6)n$ and $\text{affine-dim}(\mathcal{B}') \leq O(1/\delta)$.

The important difference between Claim 16 and [9, Corollary 1.11] is that Claim 16 guarantees the existence of a low-dimensional subspace that contains a constant fraction of the points in \mathcal{B} , whereas from [9, Corollary 1.11] we do not get any guarantee about the fraction of points from \mathcal{B} in the low-dimensional space.

3 Strengthening Case iii of Theorem 11

The following claim strengthens Theorem 11iii by providing more information on the polynomial in the radical.

▷ **Claim 17.** Let P, Q and T be irreducible homogeneous quadratic polynomials, such that $T \in \sqrt{\langle P, Q \rangle}$. Furthermore, assume that they satisfy Theorem 11iii and not any other case, that is, there are linear forms v_1, v_2 such that $T, P, Q \in \langle v_1, v_2 \rangle$. Finally, assume $\text{Lin}(P) \not\subseteq \text{Lin}(Q)$. Then there are linear forms $v'_1, v'_2 \in \text{span}\{v_1, v_2\}$ such that the following holds:

- $P = v'_1 \ell + v'^2_2$ for some linear form ℓ .
 - $Q = v'_1 u - v'^2_2$ for some linear form u .
 - $T = v'_2(\ell + u) + \alpha P + \beta Q$ for some constants $\alpha, \beta \in \mathbb{C}$,
- where the qualities holds up to a constant non zero factor.

We provide the proof of Claim 17 in the full version. As a consequence of the claim we can deduce the following uniqueness property.

▶ **Corollary 18.** Let P, Q, Q', T be pairwise linearly independent irreducible quadratics such that $T \in \sqrt{\langle P, Q \rangle}$. Let T' be such that $T' \not\sim P, Q, Q'$ and such that $T' \in \sqrt{\langle P, Q' \rangle}$. Assume further that $P \in \langle \text{Lin}(Q) + \text{Lin}(Q') \rangle$ but $\text{Lin}(P) \not\subseteq \text{Lin}(Q) + \text{Lin}(Q')$. Then $T \neq T'$. In addition, $\text{Lin}(T), \text{Lin}(T') \not\subseteq \text{Lin}(Q) + \text{Lin}(Q')$.

The proof of Corollary 18 appears in the full version.

We finish this section by formulating the improvement for Theorem 11 which follows immediately from Claim 17

▶ **Corollary 19 (Improvement of Theorem 1.10 of [40]).** Let A, B and C be n -variate, homogeneous, quadratic polynomials, over \mathbb{C} , such that $C \in \sqrt{\langle A, B \rangle}$. Then, one of the following cases must hold:

- (i) C is in the linear span of A and B .
- (ii) There exists a non trivial linear combination of the form $\alpha A + \beta B = \ell^2$ for some linear form ℓ .
- (iii) If none of the above hold, then there exist two linear forms ℓ_1 and ℓ_2 such that $A, B, C \in \langle \ell_1, \ell_2 \rangle$. Furthermore, we have that either $\text{Lin}(P) \subseteq \text{Lin}(Q)$ or
 - $A = \ell_1 a + \ell^2_2$ for some linear form a .
 - $B = \ell_1 b - \ell^2_2$ for some linear form b .
 - $C = \ell_2(a + b) + \alpha A + \beta B$ for some constants $\alpha, \beta \in \mathbb{C}$.

4 Robust Sylvester-Gallai theorem for quadratic polynomials

We divide $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \mathcal{Q}_3$ as following:

$$\mathcal{Q}_1 = \left\{ Q \in \mathcal{Q} \mid \begin{array}{l} Q \text{ satisfies Theorem 11i with at least} \\ \delta/100 \text{ fraction of the polynomials in } \mathcal{Q} \end{array} \right\}, \tag{1}$$

$$\mathcal{Q}_2 = \left\{ Q \in \mathcal{Q} \mid \begin{array}{l} Q \text{ satisfies Theorem 11ii with at least} \\ \delta/100 \text{ fraction of the polynomials in } \mathcal{Q} \end{array} \right\}, \tag{2}$$

$$\mathcal{Q}_3 = \left\{ Q \in \mathcal{Q} \mid \begin{array}{l} Q \text{ satisfies Theorem 11iii with at least} \\ \delta/100 \text{ fraction of the polynomials in } \mathcal{Q} \end{array} \right\}. \tag{3}$$

We will also use the following notation: Let $Q \in \mathcal{Q}$, and $t \in \{(i), (ii), (iii)\}$ we denote

$$\Gamma_t(Q) = \{P \in \mathcal{Q} \mid Q, P \text{ satisfy case } t \text{ of Theorem 11}\}.$$

Finally we set $\mathcal{Q}_1 = \mathcal{Q}_1 \setminus (\mathcal{Q}_2 \cup \mathcal{Q}_3)$. This implies that if $P \in \mathcal{Q}_1$ then at least a $\delta/100$ fraction of the polynomials in \mathcal{Q} satisfy Theorem 11i with P and no other case.

43:10 Robust Sylvester-Gallai Type Theorem for Quadratic Polynomials

► **Observation 20.** *The definition of Γ_t naturally defines an undirected graph with an edge between P and Q if for some t , $Q \in \Gamma_t(P)$ (which is equivalent to saying $P \in \Gamma_t(Q)$). Thus, when we speak of “edges” and “neighbors” this graph is the one that we refer to.*

Throughout the proof, we will use the following simple claim.

▷ **Claim 21.** Let $P, T \in \mathcal{Q}$. Removing T from \mathcal{Q} , causes the removal of at most two polynomials from $\Gamma_{(i)}(P)$, and this happens only in the case that $P \in \Gamma_{(i)}(T)$ and $|\mathcal{Q} \cap \text{span}\{P, T\}| = 3$.

Proof. First, note that for $Q_1, Q_2, Q_3 \in \mathcal{Q}$ if $Q_3 \in \text{span}\{Q_1, Q_2\}$, then for every $k \neq j \in [3]$, $Q_k \in \Gamma_{(i)}(Q_j)$. In particular, if $P \notin \Gamma_{(i)}(T)$, then removing T from \mathcal{Q} does not affect $\Gamma_{(i)}(P)$.

Let $P \in \Gamma_{(i)}(T)$. By the argument above, if $|\mathcal{Q} \cap \text{span}\{P, T\}| > 3$ then removing T does not affect $\Gamma_{(i)}(P)$. Thus, the only case the $\Gamma_{(i)}(P)$ is affected is when $|\mathcal{Q} \cap \text{span}\{P, T\}| = 3$ and in this case the third polynomial in the span is removed from $\Gamma_{(i)}(P)$. ◁

The proof of Theorem 8 is organized as follows. In the full version we bound the dimension of \mathcal{Q}_2 . Specifically, we prove the following claim.

▷ **Claim 22.** There exist a subset $\mathcal{I} \subseteq \mathcal{Q}_2$ of size $|\mathcal{I}| = O(1/\delta)$, and a linear space of linear forms V' such that $\dim(V') = O(1/\delta^2)$ such that $\mathcal{Q}_2 \subset \text{span}\{\mathcal{I}, \mathbb{C}[V']_2\}$.

In the full version prove that for some small dimensional space V'' , it holds that $\mathcal{Q}_3 \subset \langle V'' \rangle$.

▷ **Claim 23.** There exists a linear space of linear forms, V'' , such that $\dim(V'') = O(1/\delta)$ and $\mathcal{Q}_3 \subset \langle V'' \rangle$.

Set $V = V' + V''$. So far it holds that $\mathcal{Q}_2 \in \text{span}\{\mathcal{I}, \mathbb{C}[V]_2\}$ and $\mathcal{Q}_3 \subset \langle V \rangle$. Next, we find a small set of polynomials \mathcal{J} such that $\mathcal{Q} \subset \langle V \rangle + \text{span}\{\mathcal{J}\}$.

▷ **Claim 24.** There exists a set $\mathcal{J} \subseteq \mathcal{Q}$, of size $|\mathcal{J}| = O(1/\delta)$, such that $\mathcal{Q} \subset \text{span}\{(\mathcal{Q} \cap \langle V \rangle), \mathcal{J}, \mathbb{C}[V]_2\}$. Furthermore, if $P \in \mathcal{Q} \setminus \langle V \rangle$ then there is no quadratic L such that $P + L \in \langle V \rangle$ and $\text{rank}_s(L) \leq 2$.

Given the claims above we have that $\mathcal{Q} \subset \text{span}\{(\mathcal{Q} \cap \langle V \rangle), \mathcal{J}, \mathbb{C}[V]_2\}$, where $|\mathcal{J}| = O(1/\delta)$ and $\dim(V) = O(1/\delta^2)$. We are not done yet as the dimension of $\langle V \rangle$, as a vector space, is not a constant. To bound this dimension we partition \mathcal{Q} to four sets and study the subgraphs induced by any two of the sets.

$$\mathcal{C}_{[V]} = \{Q \in \mathcal{Q} \mid Q \in \mathbb{C}[V]_2\} \tag{4}$$

$$\mathcal{C}_{\langle V \rangle} = \{Q \in \mathcal{Q} \mid Q \in \langle V \rangle\} \setminus \mathcal{C}_{[V]} \tag{5}$$

$$\mathcal{J}_{[V]} = \{Q \in \mathcal{Q} \mid Q \in \text{span}\{\mathcal{J}, \mathbb{C}[V]_2\} \setminus \mathbb{C}[V]_2\} \tag{6}$$

$$\mathcal{J}_{\langle V \rangle} = \{Q \in \mathcal{Q} \mid Q \in \text{span}\{\mathcal{J}, \langle V \rangle\} \setminus \langle V \rangle\} \setminus \mathcal{J}_{[V]} . \tag{7}$$

In words, $\mathcal{C}_{[V]}$ is the set of all quadratics in \mathcal{Q} that only depend on linear functions in V . $\mathcal{C}_{\langle V \rangle}$ is the set of polynomials that are in $\langle V \rangle$ but not in $\mathcal{C}_{[V]}$, etc.

Our goal is to bound the dimension of each of these sets. In fact, we already know that $\dim(\mathcal{C}_{[V]})$, $\dim(\mathcal{J}_{[V]}) \leq O(1/\delta^4)$ so we only need to bound $\dim(\mathcal{C}_{\langle V \rangle})$ and $\dim(\mathcal{J}_{\langle V \rangle})$. For that we will analyze the edges between the different sets.

We first note that the “furthermore” part of Claim 24, stating that the “rank-distance” between nonzero polynomials in $\text{span}\{\mathcal{J}\}$ and quadratics in $\langle V \rangle$ is larger than 2, implies the following:

► **Observation 25.**

1. If $P \in \mathcal{C}_{\langle V \rangle} \cup \mathcal{C}_{[V]}$ and $Q \in \mathcal{J}_{\langle V \rangle} \cup \mathcal{J}_{[V]}$ satisfy that $P \in \Gamma(Q)$ then P and Q satisfy Theorem 11i.
2. If $P \in \mathcal{J}_{\langle V \rangle}$ and $Q \in \mathcal{C}_{\langle V \rangle} \cup \mathcal{C}_{[V]} \cup \mathcal{J}_{[V]}$ satisfy that $P \in \Gamma(Q)$ then P and Q satisfy Theorem 11i.

Proof. We only prove the first case as the proof of the second case is similar. As $Q \in \mathcal{J}_{\langle V \rangle} \cup \mathcal{J}_{[V]}$, we have that $\text{rank}_s(Q_1) > 2$. In particular, P and Q do not satisfy Theorem 11iii. If P and Q satisfy Theorem 11ii then $Q = \alpha P + \ell^2$ for some linear form ℓ , which contradicts the structure of \mathcal{J} guaranteed in Claim 24. ◀

To bound the dimension of $\mathcal{C}_{\langle V \rangle}$ we note that any edge going from $P \in \mathcal{C}_{\langle V \rangle} \cup \mathcal{J}_{\langle V \rangle}$ to $\mathcal{C}_{[V]} \cup \mathcal{J}_{[V]}$ defines uniquely a third polynomial in $\mathcal{C}_{\langle V \rangle} \cup \mathcal{J}_{\langle V \rangle}$. This uniqueness property guarantees that if we add $\text{Lin}(P)$ to V , then many polynomials move from $\mathcal{C}_{\langle V \rangle} \cup \mathcal{J}_{\langle V \rangle}$ to $\mathcal{C}_{[V]} \cup \mathcal{J}_{[V]}$.

► **Claim 26.** Let $P \in \mathcal{C}_{\langle V \rangle}$ then,

1. for every polynomial $Q_1 \in \Gamma(P) \cap \mathcal{J}_{[V]}$ there is a unique polynomial $Q'_1 \in \mathcal{J}_{\langle V \rangle}$ such that $Q'_1 \in \text{span}\{P, Q_1\}$. I.e., there is no other $Q_2 \in \mathcal{J}_{[V]}$ such that $Q'_1 \in \text{span}\{P, Q_2\}$.
2. for every polynomial $Q_1 \in \Gamma(P) \cap \mathcal{C}_{[V]}$ there is a unique polynomial $Q'_1 \in \mathcal{C}_{\langle V \rangle}$ such that $Q'_1 \in \sqrt{\langle P, Q_1 \rangle}$. I.e., there is no other $Q_2 \in \mathcal{C}_{[V]}$ such that $Q'_1 \in \sqrt{\langle P, Q_2 \rangle}$.

Proof.

1. Let $Q_1 \in \Gamma(P) \cap \mathcal{J}_{[V]}$. By Observation 25, P and Q_1 satisfy Theorem 11i. We first prove that they span a polynomial in $\mathcal{J}_{\langle V \rangle}$ and then prove its uniqueness. Any polynomial in $T \in \text{span}\{P, Q_1\} \setminus (\text{span}\{P\})$ has $\text{rank}_s(T) > 2$, even when setting the linear forms in V to 0. Hence, P and Q_1 span a polynomial $Q'_1 \in \mathcal{J}_{[V]} \cup \mathcal{J}_{\langle V \rangle}$. As $P \notin \mathcal{C}[V]_2$ we can conclude that $Q'_1 \in \mathcal{J}_{\langle V \rangle}$. To prove that Q'_1 is unique assume that $Q'_1 \in \text{span}\{P, Q_2\}$ for some $Q_2 \in \mathcal{J}_{[V]}$. Pairwise linear independence implies that $P \in \text{span}\{Q_1, Q_2\}$ which implies that $P \in \mathcal{C}_{[V]}$, in contradiction.
2. Follows from Corollary 18. ◀

► **Claim 27.** Let $P \in \mathcal{J}_{\langle V \rangle}$. Then for every polynomial $Q_1 \in \Gamma(P) \cap (\mathcal{J}_{[V]} \cup \mathcal{C}_{[V]})$ there is a unique polynomial $Q'_1 \in \mathcal{J}_{\langle V \rangle} \cup \mathcal{C}_{\langle V \rangle}$ such that $Q'_1 \in \text{span}\{P, Q_1\}$. By “unique” we mean that there is no other $Q_2 \in \mathcal{J}_{[V]}$ such that $Q'_1 \in \text{span}\{P, Q_2\}$.

Proof. We first consider the case $Q_1 \in \Gamma(P) \cap \mathcal{C}_{[V]}$. Observation 25 implies that P and Q_1 satisfy Theorem 11i. By construction of \mathcal{J} , any polynomial in $T \in \text{span}\{P, Q_1\} \setminus (\text{span}\{Q_1\})$ has $\text{rank}_s(T) > 2$, even when setting the linear forms in V to 0. Hence, P and Q_1 span a polynomial $Q'_1 \in \mathcal{J}_{[V]} \cup \mathcal{J}_{\langle V \rangle}$. As $P \notin \mathcal{J}_{[V]}$ we conclude that $Q'_1 \in \mathcal{J}_{\langle V \rangle}$. To prove that Q'_1 is unique assume that $Q'_1 \in \text{span}\{P, Q_2\}$ for some $Q_2 \in \mathcal{J}_{[V]} \cup \mathcal{C}_{[V]}$. As before, pairwise linear independence shows that $P \in \text{span}\{Q_1, Q_2\}$, which implies that $P \in \mathcal{J}_{[V]}$, in contradiction.

Consider the case $Q_1 \in \Gamma(P) \cap \mathcal{J}_{[V]}$. As before, P and Q_1 must satisfy Theorem 11i. Any polynomial in $T \in \text{span}\{P, Q_1\} \setminus (\text{span}\{Q_1\})$ is not in $\mathcal{J}_{[V]} \cup \mathcal{C}_{[V]}$. Hence, P and Q_1 span a polynomial $Q'_1 \in \mathcal{C}_{\langle V \rangle} \cup \mathcal{J}_{\langle V \rangle}$. Uniqueness follows exactly as in the first case. ◀

We next show that the uniqueness property proved in Claims 26 and 27 imply that $\mathcal{J}_{\langle V \rangle}$ and $\mathcal{C}_{\langle V \rangle}$ cannot be “too small,” unless they are empty.

► **Claim 28.** If $|\mathcal{J}_{\langle V \rangle}|, |\mathcal{C}_{\langle V \rangle}| \leq (\delta/10) \cdot m$, then $\mathcal{J}_{\langle V \rangle} = \mathcal{C}_{\langle V \rangle} = \emptyset$.

43:12 Robust Sylvester-Gallai Type Theorem for Quadratic Polynomials

Proof. Assume towards a contradiction that there is $P \in \mathcal{C}_{\langle V \rangle} \cup \mathcal{J}_{\langle V \rangle}$. As $|\Gamma(P)| \geq \delta m$ it follows that $|\Gamma(P) \cap (\mathcal{C}_{[V]} \cup \mathcal{J}_{[V]})| \geq (8\delta/10) \cdot m$. Claims 26 and 27 imply that there are at least $|\Gamma(P) \cap (\mathcal{C}_{[V]} \cup \mathcal{J}_{[V]})| \geq 8\delta/10$ polynomials in $\mathcal{J}_{\langle V \rangle} \cup \mathcal{C}_{\langle V \rangle}$ in contradiction to the assumption that there are at most $(2\delta/10) \cdot m$ polynomials in $\mathcal{J}_{\langle V \rangle} \cup \mathcal{C}_{\langle V \rangle}$. \triangleleft

Thus, if we can make $|\mathcal{J}_{\langle V \rangle}|, |\mathcal{C}_{\langle V \rangle}| \leq (\delta/10) \cdot m$ without increasing $\dim(V)$ and $|\mathcal{J}|$ too much then Claim 28 would imply that $\mathcal{Q} \in \text{span}\{\mathcal{J}, \mathbb{C}[V]_2\}$, from which the theorem would follow. We first show how to reduce $|\mathcal{C}_{\langle V \rangle}|$ and then we reduce $|\mathcal{J}_{\langle V \rangle}|$. We will need the following easy observation.

\triangleright **Claim 29.** There is a linear subspace $V \subseteq V'$, of dimension $\dim(V') \leq 1/\delta^4 \cdot \dim(V) \leq 1/\delta^6$, such that $|\mathcal{C}_{\langle V' \rangle}| \leq \delta/10 \cdot m$.

The proof of Claim 29 appears in the full version. Note that it may now be the case that some linear combination of polynomials in \mathcal{J} is now “close” to V' . We therefore perform the following simple process: if $Q \in \text{span}\{\mathcal{J}\}$ is such that for some quadratic L of $\text{rank}(L) = 2$ we have that $P + L \in \langle V' \rangle$ then we can add $\text{Lin}(L)$ to V' and remove one polynomial from \mathcal{J} while still maintaining that $\mathcal{Q} \subset \text{span}\{(\mathcal{Q} \cap \langle V' \rangle), \mathcal{J}, \mathbb{C}[V']_2\}$. As $|\mathcal{J}| = O(1/\delta)$, this does not have much affect on the dimension of V' , which is still $O(1/\delta^4 \cdot \dim(V))$.

To simplify notation, we denote with V the linear space guaranteed by Claim 29. As V may have changed, we update the sets $\mathcal{C}_{[V]}, \mathcal{C}_{\langle V \rangle}, \mathcal{J}_{[V]}$ and $\mathcal{J}_{\langle V \rangle}$ accordingly. By construction of $V = V'$, we now have that $|\mathcal{C}_{\langle V \rangle}| \leq \delta/100m$.

We now complete the proof of Theorem 8 by bounding the dimension of $\mathcal{J}_{\langle V \rangle}$.

\triangleright **Claim 30.** There is a set $\mathcal{J} \subseteq \mathcal{J}' \subset \mathcal{Q}$ such that $|\mathcal{J}'| \leq |\mathcal{J}| + O(1/\delta)$ and $\dim(\mathcal{J}'_{\langle V \rangle}) \leq O(1/\delta + \dim(V)^2)$.

Proof. Denote $\mathcal{T}_1 = \{Q \in \mathcal{J} \mid |\Gamma_{(ii)}(Q)| \geq 0.1\delta m\}$ and $\mathcal{T}_2 = \mathcal{J}_{\langle V \rangle} \setminus \mathcal{T}_1$. For every polynomial in $Q \in \mathcal{J}_{\langle V \rangle}$, denote $Q = Q_{\mathcal{J}} + Q_{\langle V \rangle}$ where $Q_{\mathcal{J}} \in \text{span}\{\mathcal{J}\}$ and $Q_{\langle V \rangle} \in \langle V \rangle$. Note that neither $Q_{\mathcal{J}}$ nor $Q_{\langle V \rangle}$ can be zero as this would imply $Q \in \mathcal{J}_{[V]} \cup \mathcal{C}_{\langle V \rangle}$.

\triangleright **Claim 31.** There is a subset $\mathcal{T}'_1 \subseteq \mathcal{T}_1$ of size at most $10/\delta$ such that $\mathcal{T}_1 \subset \text{span}\{\mathcal{T}'_1, \mathcal{J}, \mathbb{C}[V]_2\}$.

We prove Claim 31 in the full version. Set $\mathcal{T}_2 = \mathcal{T}_2 \setminus \text{span}\{\mathcal{T}_1, \mathcal{J}, \mathbb{C}[V]_2\}$. Every $Q \in \mathcal{T}_2$ must now satisfy that $|\Gamma_i(Q)| \geq 0.9\delta m$. Indeed, this follows from the fact that $Q \notin \mathcal{T}_1$ and that it cannot satisfy Theorem 11iii with any polynomial. Remove from $\Gamma_i(Q)$ all the polynomials in \mathcal{B}_1 , this removes at most $2|\mathcal{B}_1| \leq 2/10\delta m$ polynomials from $\Gamma_i(Q)$ (using an argument similar to Claim 21), leaving $|\Gamma_{(i)}(Q)| \geq 0.7\delta m$. This implies that $\mathcal{K} = \mathcal{T}_2$, $W = \text{span}\{\mathcal{T}_1, \mathcal{J}, \mathbb{C}[V]_2\}$ and $\mathcal{W} = \mathcal{Q} \cap \text{span}\{\mathcal{T}_1, \mathcal{J}, \mathbb{C}[V]_2\}$ satisfy the conditions of Theorem 15. As $\dim(W) \leq O(\dim(V)^2)$ it follows that $\dim(\mathcal{J}_{\langle V \rangle}) \leq O(1/\delta + \dim(V)^2)$.

Setting $\mathcal{J}' = \mathcal{T}'_1 \cup \mathcal{J}$ completes the proof. \triangleleft

We now put everything together and prove Theorem 8.

Proof of Theorem 8. Claims 22, 23 and 24 imply that there exists a set $\mathcal{J} \subseteq \mathcal{Q}$, of size $|\mathcal{J}| = O(1/\delta)$, and a subspace of linear functions V of dimension $\dim(V) = O(1/\delta^2)$ such that $\mathcal{Q} \subset \text{span}\{(\mathcal{Q} \cap \langle V \rangle), \mathcal{J}, \mathbb{C}[V]_2\}$.

By Claims 29 and 30 there are $\mathcal{J} \subseteq \mathcal{J}'$ and $V \subseteq V'$ such that $\dim(V') \leq 1/\delta^4 \cdot \dim(V) \leq 1/\delta^6$ and $|\mathcal{J}'| = O(1/\delta)$, for which it holds that $|\mathcal{C}_{\langle V' \rangle}| \leq \delta/10 \cdot m$ and $\dim(\mathcal{J}'_{\langle V \rangle}) \leq O(1/\delta + \dim(V)^2) = O(1/\delta^8)$. We now set $\mathcal{J} = \mathcal{J}'$, $V = V'$ and, if needed, we add $O(|\mathcal{J}|)$ linear functions to V to make sure that no non-trivial linear combination of polynomials in

\mathcal{J} is of the form $L + F(V)$ where $\text{rank}_s(L) \leq 2$ and $F \in \mathbb{C}[V]_2$, we obtain that $\mathcal{J}_{\langle V \rangle} = \emptyset$ and $|\mathcal{C}_{\langle V \rangle}| \leq \delta m/10$. Claim 28 now guarantees that we also have that $\mathcal{C}_{\langle V \rangle} = \emptyset$. Hence, $\mathcal{Q} = \mathcal{C}_{[V]} \cup \mathcal{J}_{[V]}$ and it follows that $\dim(\text{span}\{\mathcal{Q}\}) \leq |\mathcal{J}| + \dim(V)^2 = O(1/\delta^{16})$. \blacktriangleleft

References

- 1 Manindra Agrawal. Proving lower bounds via pseudo-random generators. In Ramaswamy Ramanujam and Sandeep Sen, editors, *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science, 25th International Conference, Hyderabad, India, December 15-18, 2005, Proceedings*, volume 3821 of *Lecture Notes in Computer Science*, pages 92–105. Springer, 2005. doi:10.1007/11590156_6.
- 2 Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 67–75. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.32.
- 3 Boaz Barak, Zeev Dvir, Avi Wigderson, and Amir Yehudayoff. Fractional Sylvester–Gallai theorems. *Proceedings of the National Academy of Sciences*, 110(48):19213–19219, 2013.
- 4 Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic independence and blackbox identity testing. *Inf. Comput.*, 222:2–19, 2013. doi:10.1016/j.ic.2012.10.004.
- 5 Peter Borwein and William O. J. Moser. A survey of Sylvester’s problem and its generalizations. *Aequationes Mathematicae*, 40:111–135, 1990. doi:10.1007/BF02112289.
- 6 Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Hardness vs Randomness for Bounded Depth Arithmetic Circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPICs*, pages 13:1–13:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.CCC.2018.13.
- 7 Zeev Dvir. Incidence theorems and their applications. *Found. Trends Theor. Comput. Sci.*, 6(4):257–393, 2012. doi:10.1561/04000000056.
- 8 Zeev Dvir and Guangda Hu. Sylvester–Gallai for Arrangements of Subspaces. *Discrete & Computational Geometry*, 56(4):940–965, 2016. doi:10.1007/s00454-016-9781-7.
- 9 Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Improved rank bounds for design matrices and a new proof of Kelly’s theorem. *Forum of Mathematics, Sigma*, 2, 2014. arXiv:1211.0330.
- 10 Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM J. Comput.*, 36(5):1404–1434, 2007. doi:10.1137/05063605X.
- 11 Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009. doi:10.1137/080735850.
- 12 Paul Erdős. Problems for Solution: 4065. *The American Mathematical Monthly*, 50(1):65, 1943. URL: <http://www.jstor.org/stable/2304011>.
- 13 Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. A deterministic parallel algorithm for bipartite perfect matching. *Commun. ACM*, 62(3):109–115, 2019. doi:10.1145/3306208.
- 14 Michael A. Forbes. *Polynomial identity testing of read-once oblivious algebraic branching programs*. PhD thesis, Massachusetts Institute of Technology, 2014.
- 15 Michael A. Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In Prasad Raghavendra, Sofya Raskhodnikova, Klaus Jansen, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, volume 8096 of *Lecture Notes in Computer Science*, pages 527–542. Springer, 2013. doi:10.1007/978-3-642-40328-6_37.

- 16 Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct Hitting Sets and Barriers to Proving Lower Bounds for Algebraic Circuits. *Theory of Computing*, 14(1):1–45, 2018. doi:10.4086/toc.2018.v014a018.
- 17 Tibor Gallai. Solution to Problem 4065. *The American Mathematical Monthly*, 51:169–171, 1944.
- 18 Abhibhav Garg, Rafael Oliveira, and Akash Kumar Sengupta. Robust Radical Sylvester-Gallai Theorem for Quadratics. Personal communication, 2021.
- 19 Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017. arXiv:1701.01717.
- 20 Ankit Gupta. Algebraic Geometric Techniques for Depth-4 PIT & Sylvester-Gallai Conjectures for Varieties. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:130, 2014. URL: <http://eccc.hpi-web.de/report/2014/130>.
- 21 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. doi:10.1137/140957123.
- 22 Sten Hansen. A generalization of a theorem of Sylvester on the lines determined by a finite point set. *Mathematica Scandinavica*, 16:175–180, 1965.
- 23 Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute (extended abstract). In Raymond E. Miller, Seymour Ginsburg, Walter A. Burkhard, and Richard J. Lipton, editors, *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 262–272. ACM, 1980. doi:10.1145/800141.804674.
- 24 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. doi:10.1007/s00037-004-0182-6.
- 25 Zohar S. Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 274–285. IEEE Computer Society, 2009. doi:10.1109/CCC.2009.18.
- 26 Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 198–207. IEEE Computer Society, 2009. doi:10.1109/FOCS.2009.67.
- 27 Leroy Milton Kelly. A resolution of the Sylvester-Gallai problem of J.-P. Serre. *Discrete & Computational Geometry*, 1(2):101–104, 1986.
- 28 Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and polynomial factorization. *Computational Complexity*, 24(2):295–331, 2015. doi:10.1007/s00037-015-0102-y.
- 29 Mrinal Kumar and Ramprasad Saptharishi. Hardness-Randomness Tradeoffs for Algebraic Computation. *Bull. EATCS*, 129, 2019. URL: <http://bulletin.eatcs.org/index.php/beatcs/article/view/591/599>.
- 30 Eberhard Melchior. Über Vielseite der Projektive Ebene. *Deutsche Mathematik*, 5:461–475, 1941.
- 31 Ketan D. Mulmuley. Geometric complexity theory V: Efficient algorithms for Noether normalization. *J. Amer. Math. Soc.*, 30(1):225–309, 2017.
- 32 Shir Peleg and Amir Shpilka. A Generalized Sylvester-Gallai Type Theorem for Quadratic Polynomials. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPICs*, pages 8:1–8:33. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.CCC.2020.8.

- 33 Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits via edelstein-kelly type theorem for quadratic polynomials. *CoRR*, abs/2006.08263, 2020. [arXiv:2006.08263](https://arxiv.org/abs/2006.08263).
- 34 Nitin Saxena. Progress on polynomial identity testing. *Bulletin of EATCS*, 99:49–79, 2009. URL: <https://eccc.weizmann.ac.il/report/2009/101/>.
- 35 Nitin Saxena. Progress on polynomial identity testing-ii. In M. Agrawal and V. Arvind, editors, *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*, Progress in Computer Science and Applied Logic, pages 131–146. Springer International Publishing, 2014. URL: <https://books.google.co.il/books?id=U7ApBAAAQBAJ>.
- 36 Nitin Saxena and Comandur Seshadhri. Blackbox identity testing for bounded top-fan-in depth-3 circuits: The field doesn't matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012. doi:10.1137/10848232.
- 37 Nitin Saxena and Comandur Seshadhri. From sylvester-gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *J. ACM*, 60(5):33, 2013. doi:10.1145/2528403.
- 38 Jean-Pierre Serre. Advanced Problems: 5359. *The American Mathematical Monthly*, 73(1):89, 1966. URL: <http://www.jstor.org/stable/2313941>.
- 39 Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. *SIAM J. Comput.*, 38(6):2130–2161, 2009. doi:10.1137/070694879.
- 40 Amir Shpilka. Sylvester-Gallai type theorems for quadratic polynomials. *Discrete Analysis*, 13, 2020.
- 41 Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010. doi:10.1561/04000000039.
- 42 Gaurav Sinha. Reconstruction of Real Depth-3 Circuits with Top Fan-In 2. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPICs*, pages 31:1–31:53. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.CCC.2016.31.
- 43 Ola Svensson and Jakub Tarnawski. The Matching Problem in General Graphs Is in Quasi-NC. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.70.
- 44 James Joseph Sylvester. Mathematical question 11851. *Educational Times*, pages 59–98, 1893.

Swap, Shift and Trim to Edge Collapse a Filtration

Marc Glisse   

Université Paris-Saclay, CNRS, Inria, Laboratoire de Mathématiques d'Orsay, 91405, Orsay, France

Siddharth Pritam  

School of Engineering, Department of Computer Science, Shiv Nadar University, Delhi NCR, India

Abstract

Boissonnat and Pritam introduced an algorithm to reduce a filtration of flag (or clique) complexes, which can in particular speed up the computation of its persistent homology. They used so-called *edge collapse* to reduce the input flag filtration and their reduction method required only the 1-skeleton of the filtration. In this paper we revisit the use of edge collapse for efficient computation of persistent homology. We first give a simple and intuitive explanation of the principles underlying that algorithm. This in turn allows us to propose various extensions including a zigzag filtration simplification algorithm. We finally show some experiments to better understand how it behaves.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Algebraic topology

Keywords and phrases edge collapse, flag complex, graph, persistent homology

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.44

Related Version *Full Version*: <https://arxiv.org/abs/2203.07022>

Supplementary Material *Software (Source Code)*: <https://github.com/GUDHI/gudhi-devel> [16]
archived at `swh:1:cnt:c823901feab91f79f85da1717314127803fe18fd`

1 Introduction

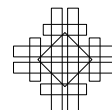
Efficient computation of persistent homology has been a central quest in Topological Data Analysis (TDA) since the early days of the field about 20 years ago. Given a filtration (a nested sequence of simplicial complexes), computation of persistent homology involves reduction of a boundary matrix, whose rows and columns are the simplices of the input filtration. Traditionally, there are two complementary lines of research that have been explored to improve the computation of persistent homology. The first approach led to improvement of the persistence algorithm (the boundary matrix reduction algorithm) and of its analysis, to efficient implementations and optimizations, and to a new generation of software [16, 4, 3, 18, 22, 26, 1]. The second and complementary approach is to reduce (or simplify) the input filtration to a smaller filtration through various geometric or topological techniques in an exact or approximate way and then compute the persistent homology of the smaller reduced filtration. This research direction has been intensively explored as well [21, 14, 10, 7, 27, 19, 11, 13].

Flag complexes and, in particular, the Vietoris-Rips complexes are an important class of simplicial complexes that are extensively used in TDA. Flag complexes are fully characterized by their graph (or 1-skeleton) and can thus be stored in a very compact way. Therefore, they are of great practical importance and are well studied theoretically. Various efficient codes and reduction techniques have been developed for those complexes [3, 27, 26, 1]. However, further progress have been made only recently by the work of Boissonnat and Pritam [5, 6]. Both works [5, 6] put forward preprocessing techniques, which reduce an input flag filtration (nested sequence of flag complexes) to a smaller flag filtration using only the 1-skeleton. The work in [5] uses a special type of collapse called strong collapse (removal of special vertices called dominated vertices), introduced by J. Barmak and E. Miniam [2]. In [6] they extend



© Marc Glisse and Siddharth Pritam;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 44; pp. 44:1–44:15
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



the notion of strong collapse to edge collapse (removal of special edges, called dominated edges) and use it for further filtration simplification which improves the performance by several orders of magnitude.

In this paper, we revisit the usage of edge collapse for efficient computation of persistent homology. We first give a simple and intuitive explanation of the principles underlying the algorithm proposed in [6]. We identify that an algorithm to edge collapse a filtration can be deconstructed as three fundamental operations: 1. *Swap* two edges having same filtration value, 2. *Shift* a dominated edge forward in the filtration and 3. *Trim* the very last dominated edge. This new approach allows us to propose various extensions, which we list below.

- **Backward:** We propose a backward reduction algorithm, which processes the edges of a flag filtration with decreasing filtration values. The algorithm in [6] processes edges one by one with increasing filtration values, i.e. in the forward direction. The backward processing results (shown experimentally) in faster reduction of the edges as it allows various operations like domination checks, computing the neighbourhood of an edge etc to be performed fewer times than in the forward algorithm of [6].
- **Parallel:** We propose a divide and conquer heuristic to further improve and semi-parallelize our backward reduction algorithm. Our approach is to subdivide the input filtration into two smaller sub-sequences (consisting of consecutive edges), we process these smaller sub-sequences in parallel and then merge the solutions of two sequences to form the solution of the complete sequence. The two sub-sequences can be further sub-divided and processed recursively in parallel.
- **Approximate:** With this simplified perspective a simple tweak in the backward algorithm allows us to have an approximate version of the reduction algorithm. There are two goals in mind behind an approximate version, first to speed up the algorithm, and second to obtain a smaller reduced sequence. We perform certain experiments to show how the approximate version performs on these two parameters.
- **Zigzag:** We provide a (parallelizable) reduction algorithm for a zigzag flag filtration, which is a sequence of flag complexes linked through inclusion maps in both forward and backward directions.

We note that we don't assume that all the vertices appear in the beginning of the filtration. That is the filtration values of vertices can be arbitrary as well.

2 Background

We briefly recall the basic notions like simplicial complexes, flag complexes, persistent homology and edge collapse. For more details on these topics please refer to [15, 17, 23].

Simplicial complex and simplicial map. An **abstract simplicial complex** K is a collection of subsets of a non-empty finite set X , such that for every subset A in K , all the subsets of A are in K . We call an *abstract simplicial complex* simply a *simplicial complex* or just a *complex*. An element of K is called a **simplex**. An element of cardinality $k + 1$ is called a k -simplex and k is called its **dimension**. Given a simplicial complex K , we denote its geometric realization as $|K|$. A simplex is called **maximal** if it is not a proper subset of any other simplex in K . A sub-collection L of K is called a **subcomplex** if it is a simplicial complex itself. An inclusion $\psi : K \xrightarrow{\sigma} K \cup \sigma$ of a single simplex σ is called **elementary**, otherwise, it's called **non-elementary**. An inclusion $\psi : K \hookrightarrow L$ between two complexes K and L induces a continuous map $|\psi| : |K| \rightarrow |L|$ between the underlying geometric realizations.

Flag complex and neighborhood. A complex K is a **flag** or a **clique** complex if, when a subset of its vertices forms a clique (i.e. any pair of vertices is joined by an edge), they span a simplex. It follows that the full structure of K is determined by its 1-skeleton (or graph) we denote by G . For a vertex v in G , the **open neighborhood** $N_G(v)$ of v in G is defined as $N_G(v) := \{u \in G \mid [uv] \in E\}$, where E is the set of edges of G . The **closed neighborhood** $N_G[v]$ is $N_G[v] := N_G(v) \cup \{v\}$. Similarly we define the closed and open neighborhood of an edge $[xy] \in E$, $N_G[xy]$ and $N_G(xy)$ as $N_G[xy] := N_G[x] \cap N_G[y]$ and $N_G(xy) := N_G(x) \cap N_G(y)$, respectively.

Persistent homology. A **sequence** of simplicial complexes $\mathcal{F} : \{K_1 \hookrightarrow K_2 \hookrightarrow \dots \hookrightarrow K_m\}$ connected through inclusion maps is called a **filtration**. A filtration is a **flag filtration** if all the simplicial complexes K_i are flag complexes.

If we compute the homology groups of all the K_i , we get the sequence $\mathcal{P}(\mathcal{F}) : \{H_p(K_1) \xrightarrow{*} H_p(K_2) \xrightarrow{*} \dots \xrightarrow{*} H_p(K_m)\}$. Here $H_p()$ denotes the homology group of dimension p with coefficients from a field \mathbb{F} and $\xrightarrow{*}$ is the homomorphism induced by the inclusion map. $\mathcal{P}(\mathcal{F})$ is a sequence of vector spaces connected through the homomorphisms and it is called a **persistence module**. More formally, a *persistence module* \mathbb{V} is a sequence of vector spaces $\{V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow \dots \rightarrow V_m\}$ connected with homomorphisms $\{\rightarrow\}$. A persistence module arising from a filtration captures the evolution of the topology of the sequence.

Any persistence module can be *decomposed* into a collection of intervals of the form $[i, j]$ [9]. The multiset of all the intervals $[i, j]$ in this decomposition is called the **persistence diagram** (PD) of the persistence module. An interval of the form $[i, j]$ corresponds to a homological feature (a “cycle”) which appeared at i and disappeared at j . The PD completely characterizes the persistence module, that is, there is a bijective correspondence between the PD and the equivalence class of the persistence module [15, 28].

Two different persistence modules $\mathbb{V} : \{V_1 \rightarrow V_2 \rightarrow \dots \rightarrow V_m\}$ and $\mathbb{W} : \{W_1 \rightarrow W_2 \rightarrow \dots \rightarrow W_m\}$, connected through a set of homomorphisms $\phi_i : V_i \rightarrow W_i$ are **equivalent** if the ϕ_i are isomorphisms and the following diagram commutes [15, 12]. Equivalent persistence modules have the same interval decomposition, hence the same diagram.

$$\begin{array}{ccccccc}
 V_1 & \longrightarrow & V_2 & \longrightarrow & \dots & \longrightarrow & V_{m-1} & \longrightarrow & V_m \\
 \downarrow \phi_1 & & \downarrow \phi_2 & & & & \downarrow \phi_{m-1} & & \downarrow \phi_m \\
 W_1 & \longrightarrow & W_2 & \longrightarrow & \dots & \longrightarrow & W_{m-1} & \longrightarrow & W_m
 \end{array}$$

Edge collapse of a flag complex. In a flag complex K , we say that an edge $e = [ab]$, connecting vertices a and b , is **dominated** by a vertex v (different from a and b) if $N_G[e] \subseteq N_G[v]$. Removing e and all its cofaces from K defines a smaller flag complex K' . It has been proven in [6] that when e is dominated in K , the inclusion $K' \subset K$ induces an isomorphism between the homology groups of K' and K . This removal is called an **edge collapse**.

3 Swapping, shifting and trimming

In this Section, we show three simple and fundamental operations that preserve the persistence diagram of a flag filtration: 1. Swapping two edges with the same filtration value, 2. Shifting a dominated edge, and 3. Trimming a dominated edge at the end of the filtration. These operations can be combined to simplify a flag filtration.

Before we proceed, we will fix some notations. Let $\{t_1, t_2, \dots, t_n\}$ be a finite index set where $t_i \in \mathbb{R}$ and $t_i < t_j$ for $i < j$. For convenience, we may consider $t_{n+1} = \infty$. With each t_i (called the *filtration value* or *grade*) we associate a graph G_{t_i} such that $G_{t_i} \hookrightarrow G_{t_{i+1}}$ is

44:4 Swap, Shift and Trim to Edge Collapse a Filtration

an *inclusion*, (not necessarily elementary) of edges. The flag complex of G_{t_i} is denoted as \overline{G}_{t_i} and we consider the associated flag filtration $\mathcal{F} : \overline{G}_{t_1} \hookrightarrow \overline{G}_{t_2} \hookrightarrow \dots \hookrightarrow \overline{G}_{t_n}$. The edges in the set $E := \{e_1, e_2, \dots, e_m\}$ ($m \geq n$) are thus indexed with an order compatible with the filtration values.

Swapping. Inserting several edges at the same filtration value can be done in any order. We state this basic observation as the following lemma.

► **Lemma 1** (Swapping Lemma). *Given a flag filtration $\{\overline{G}_{t_1} \hookrightarrow \overline{G}_{t_2} \hookrightarrow \dots \hookrightarrow \overline{G}_{t_n}\}$, such that $G_{t_i} \hookrightarrow G_{t_{i+1}}$ is a non-elementary inclusion. Then, the indices of the edges $G_{t_{i+1}} \setminus G_{t_i}$ could be assigned interchangeably. That is, swapping their order of insertion preserves the persistence diagram.*

Shifting. In a filtration, insertion of a dominated edge does not bring immediate topological change. Therefore, its insertion can be shifted until the next grade and possibly even further.

► **Lemma 2** (Shifting Lemma). *Let e be a dominated edge in G_{t_i} inserted at grade t_i . Then, the insertion of e can be shifted by one grade to t_{i+1} without changing the persistence diagram. In other words, the persistence diagrams of the original flag filtration $\mathcal{F} := \{\overline{G}_{t_1} \hookrightarrow \dots \hookrightarrow \overline{G}_{t_i} \hookrightarrow \overline{G}_{t_{i+1}} \hookrightarrow \dots \hookrightarrow \overline{G}_{t_n}\}$ and the shifted filtration $\{\overline{G}_{t_1} \hookrightarrow \dots \hookrightarrow \overline{G_{t_i} \setminus e} \hookrightarrow \overline{G_{t_{i+1}}} \hookrightarrow \dots \hookrightarrow \overline{G_{t_n}}\}$ are equivalent.*

Proof. The proof follows from the commutativity of the following diagram, where all maps are induced by inclusions, and the fact that all vertical maps are isomorphisms.

$$\begin{array}{ccccc} H_p(\overline{G_{t_{i-1}}}) & \hookrightarrow & H_p(\overline{G_{t_i}}) & \hookrightarrow & H_p(\overline{G_{t_{i+1}}}) \\ \parallel & & \updownarrow |r_i|_* & & \parallel \\ H_p(\overline{G_{t_{i-1}}}) & \hookrightarrow & H_p(\overline{G_{t_i} \setminus e}) & \hookrightarrow & H_p(\overline{G_{t_{i+1}}}) \end{array}$$

This implies that the persistence diagrams of the sequences $\{\overline{G}_{t_1} \hookrightarrow \dots \hookrightarrow \overline{G_{t_i}} \hookrightarrow \overline{G_{t_{i+1}}} \hookrightarrow \dots \hookrightarrow \overline{G_{t_n}}\}$ and $\{\overline{G_{t_1}} \hookrightarrow \dots \hookrightarrow \overline{G_{t_i} \setminus e} \hookrightarrow \overline{G_{t_{i+1}}} \hookrightarrow \dots \hookrightarrow \overline{G_{t_n}}\}$ are equivalent, see [6, Theorem 4] for more details. Here, $|r_i|_*$ is the isomorphism between the homology groups induced by the retraction map (on the geometric realizations of the complexes) associated to the edge collapse. ◀

After an edge is shifted to grade t_{i+1} , it can leap frog the edges inserted at grade t_{i+1} using the swapping lemma (Lemma 1) and can be checked for shifting to the next grade.

Trimming. If the very last edge in the filtration is dominated then we can omit its inclusion. This is a special case of the shifting operation (Lemma 2) assuming that there is a graph G_∞ at infinity.

► **Lemma 3** (Trimming Lemma). *Let $e \notin G_{t_{n-1}}$ be a dominated edge in the graph G_{t_n} . Then, the persistence diagrams of the original sequence $\mathcal{F} := \{\overline{G}_{t_1} \hookrightarrow \overline{G}_{t_2} \hookrightarrow \dots \hookrightarrow \overline{G}_{t_n}\}$ and the trimmed sequence $\{\overline{G}_{t_1} \hookrightarrow \overline{G}_{t_2} \hookrightarrow \dots \hookrightarrow \overline{G_{t_n} \setminus e}\}$ are equivalent.*

Note that when shifting or trimming produces a sequence with identical consecutive graphs $G_{t_i} = G_{t_{i+1}}$, we can just drop index t_{i+1} .

► **Lemma 4** (Adjacency). *Let e be an edge in a graph G and let e' be a new edge with $G' := G \cup e'$. If $N_G(e) = N_{G'}(e)$ and e is dominated in G , then e is also dominated in G' .*

This is in particular the case if e and e' are not boundary edges of a common triangle in $\overline{G'}$. The above lemma is not strictly necessary, but it is useful to speed up algorithms.

4 Persistence simplification

In this Section, we will describe our new approach to use edge collapse to speed up the persistence computation. As mentioned before, the simplification process will be seen as a combination of the basic operations described in Section 3. This new perspective simplifies the design process and correctness proof of the algorithm. Along with this we achieve a significant improvement in the run-time efficiency as shown in Section 8. We first briefly look at the forward algorithm of [6] with this new point of view and then present the new approach called the *backward algorithm* [Algorithm 1]. Both algorithms take as input a flag filtration \mathcal{F} represented as a sorted array E of edges (pairs of vertices) with their filtration value, and output a similar array E^c , sorted in the case the Forward Algorithm and unsorted for the Backward Algorithm, that represents a reduced filtration \mathcal{F}^c that has the same persistence diagram as \mathcal{F} .

Forward algorithm. In the forward algorithm (the original one from [6]), the edges are processed in the order of the filtration in a streaming fashion. If a new edge is dominated, we skip its insertion and consider the next edge. If the next edge is dominated as well its insertion is skipped as well. Intuitively, the sequence of such dominated edges forms a train of dominated edges that we are moving to the right. When a new edge e is non-dominated (called *critical*), we output it, and also check what part of the train of dominated edges is allowed to continue to the right (shifted forward) and what part has to stop right there. For all the previously dominated edges (actually only those that are *adjacent* to e), we check if they are still dominated after adding the edge e . If an edge e' becomes critical, we output it with the same filtration value as e , and the following edges now have to cross both e and e' to remain in the train. We stop after processing the last edge, and the edges that are still part of the dominated train are discarded (trimmed).

Backward algorithm. The backward algorithm (Algorithm 1) considers edges in order of decreasing filtration value. Each edge e is considered once, delayed (shifted) as much as possible, then never touched again. We always implicitly swap edges so that while e is the edge considered, it is the last one inserted at its current filtration value, and compute its domination status there. If the edge is dominated, we shift it to the next filtration value, and iterate, swapping and checking for domination again at this new filtration value. If there is no next filtration value, we remove the edge (trimming). Once the edge is not dominated, we update its filtration value and output it. As an optimization, instead of moving the edge one grade at a time, we may jump directly to the filtration value of the next adjacent edge, since we know that moving across the other edges will preserve the domination (Lemma 4).

The main datastructure used here is a neighborhood map N . For each vertex u , it provides another map $N[u]$ from the adjacent vertices v_i to the filtration value $N[u][v_i]$ of edge uv_i . The two main uses of this map are computing the neighborhood of an edge uv at a time t (i.e. in the graph G_t) as $N_t[uv] = N_t[u] \cap N_t[v]$ (filtering out the edges of filtration value larger than t), and checking if such an edge neighborhood is included in the neighborhood of a vertex w at time t . While computing $N_t[uv]$, we also get as a side product the list of the future neighbors $F_t[uv]$, which we sort by filtration value. These operations can be done efficiently by keeping the maps sorted, or using hashtables. The information in N is symmetric, any operation we mention on $N[u][v]$ (removal or updating t) will also implicitly be done on $N[v][u]$. In this Section, we denote $t(e)$ the filtration value of $e \in E$, which is stored as $N[u][v]$ if $e = uv$. Note that even though E is sorted, since several edges may have the same filtration value, $G_{t(e)}$ may contain some edges that appear after e .

44:6 Swap, Shift and Trim to Edge Collapse a Filtration

We now explain the precise computation of the reduced sequence of edges E^c . See Algorithm 1 for the pseudo-code. The main **for** loop on line 4 (called the backward loop) iterates over the edges in the sequence E by decreasing filtration values, i.e. in the *backward direction*, and checks whether or not the current edge e is dominated in the graph $G_{t(e)}$. If *not*, we insert e in E^c and keep its original filtration value $t(e)$. Else, e is dominated in $G_{t(e)}$, and we increase $t(e)$ to the smallest value $t' > t(e)$ where $N_{t(e)}[e] \subsetneq N_{t'}[e]$. We can then iterate (**goto** on line 12), check if the edge is still dominated at its new filtration value t' , etc. When the edge stops being dominated, we insert it in E^c with its new $t(e)$ and update $t(e)$ in the neighborhood map N . If the smallest value $t' > t(e)$ does not actually exist, we remove the edge from the neighborhood map and do *not* insert it in E^c .

■ **Algorithm 1** Core flag filtration backward algorithm.

```

1: procedure CORE-FLAG-FILTRATION( $E$ )
2:   input : set of edges  $E$  sorted by filtration value
3:    $E^c \leftarrow \emptyset$ 
4:   for  $e \in E$  do                                     ▷ In non-increasing order of  $t(e)$ 
5:     Compute  $N_{t(e)}(e)$  and  $F_{t(e)}(e)$ 
6:     for  $w \in N_{t(e)}(e)$  do
7:       Test if  $w$  dominates  $e$  at  $t(e)$ 
8:     end for
9:     if  $e$  is dominated in  $G_{t(e)}$  then
10:      if  $F_{t(e)}(e)$  is empty then
11:        Remove  $N[u][v]$                                ▷ Trimming.
12:        go to 23 (next edge)
13:      else                                             ▷ Shift and Swap.
14:         $t' \leftarrow$  filtration of the first element of  $F_{t(e)}(e)$ 
15:        Move from  $F_{t(e)}(e)$  to  $N_{t(e)}(e)$  the vertices that become neighbors of  $e$  at  $t'$ 
16:         $N[u][v] = t(e) \leftarrow t'$ 
17:        go to 6
18:      end if
19:    else
20:      Insert  $\{e, t(e)\}$  in  $E^c$ 
21:      go to 23 (next edge)
22:    end if
23:  end for
24:  return  $E^c$                                          ▷  $E^c$  is the 1-skeleton of the core flag filtration.
25: end procedure

```

► **Theorem 5** (Correctness). *Let \mathcal{F} be a flag filtration, and \mathcal{F}^c the reduced filtration produced by Algorithm 1. \mathcal{F} and \mathcal{F}^c have the same persistence diagram.*

Proof. The proof is based on the observation that the algorithm inductively performs the elementary operations from Section 3: either it trims the very last edge of the current sequence (Line 11) or shifts and swaps a dominated edge forward to get a new sequence. Then the result follows using Lemmas 1–3 inductively. The only subtlety is around Line 15, where instead of simply performing one shift to the next filtration value, we perform a whole sequence of operations. We first shift e to the next filtration value t' (and implicitly swap e with the other edges of filtration value t'). As long as we have not reached the first element of

$F_{t(e)}(e)$, we know that shifting has not changed the neighborhood of e and thus by Lemma 4 the fact that e is dominated. We can then safely keep shifting (and swapping) until we reach that first element of $F_{t(e)}(e)$. ◀

Complexity. We write n_e for the total number of edges and k for the maximum degree of a vertex in G_{t_n} . The main loop of the procedure, Line 4 of Algorithm 1, is executed n_e times. Nested, we loop (in the form of `go to 6`) on the elements of $F_{t(e)}(e)$, of which there are at most k . For each of those elements, on Line 6, we iterate on $N_{t(e)}(e)$, which has size at most k . Finally, testing if a specific vertex dominates a specific edge amounts to checking if one set is included in another, which takes linear time in k for sorted sets or hash tables. The other operations are comparatively of negligible cost. Sorting $F_{t(e)}(e)$ on Line 5 takes time $k \log k = o(k^2)$. Line 15 may take time $k \log k$ depending on the datastructure, $O(k^2)$ in any case. This yields a complexity of $O(n_e k^3)$.

5 Parallelisation

Delaying the insertion of an edge until the next grade, and possibly swapping it, is a very local operation. As such, there is no problem doing several of them in parallel as long as they are in disjoint intervals of filtration values. We exploit this observation and further optimize our algorithm by parallelizing a significant part of the computation using a divide and conquer approach.

To describe the parallel approach, let us use the same notations t_i , G_{t_i} , \mathcal{F} , $G_{\mathcal{F}}$ and E as in Section 3. To make things simpler, we assume that all edges have distinct filtration values. We subdivide the given input edge set $E := \{e_1, e_2, \dots, e_n\}$ of size n into two smaller halves: the left half $E_l := \{e_1, e_2, \dots, e_{n/2}\}$ and the right half $E_r := \{e_{n/2+1}, e_{n/2+2}, \dots, e_n\}$ of roughly the same size. We will describe a version of the algorithm based on the backward algorithm, but the same could be done with the forward algorithm, or a mixture of both.

We first apply the backward algorithm to E_l normally (*left call*), which produces a reduced E_l^c . We also remember the list of all edges that were removed in this procedure: $E_l^r := E_l \setminus E_l^c$. Independently (in parallel), we apply the backward algorithm to E (*right call*), but stop after processing all the edges of E_r on Line 4 of Algorithm 1. In a final sequential merging step, we resume the right call, processing only the edges of E_l^r , as if they all had the same initial filtration value $t_{n/2+1}$. The subdivision can obviously be applied recursively to increase the parallelism.

► **Lemma 6.** *The parallel algorithm produces exactly the same output as the sequential algorithm, and is thus correct.*

Proof. The right call and the sequential algorithm start by handling the edges of E_r in exactly the same way. When we reach the edges of E_l , for each edge e , there are two cases. Either the sequential algorithm shifts e no further than $t_{n/2}$, in which case the left call does the same. Or the sequential algorithm shifts e further (possibly all the way to removing it), then shifting to $t_{n/2+1}$ is handled by the left call, while the rest of the shift happens in the merging step. ◀

6 Approximation

Another interesting extension is an approximate version that gives a diagram within bottleneck distance ϵ of the true diagram (or some other similar criterion). Since the Rips filtration is often used as an approximation of the Čech filtration, an additional error is often acceptable.

If an edge is non-dominated for a short range of filtration values and becomes dominated again afterwards, it is tempting to skip the non-dominated region and keep delaying this edge. However, if we are not careful, the errors caused by these skips may add up and result in a diagram that is far from the original. The simplest idea would be to round all filtration values to the nearest multiple of ϵ before running the backward algorithm (as in [5]). However, we can do a little better.

We describe here one safe approximation algorithm, based on the backward algorithm. When considering a new edge e , instead of checking if it is dominated at its original position $t(e)$, we start checking ϵ later, at filtration value $t(e) + \epsilon$. If it is dominated, we resume normal processing from there. However, if the edge is not dominated ϵ after its original insertion time, we keep it at its original position and avoid uselessly shifting the whole sequence.

► **Lemma 7.** *The resulting module is ϵ -interleaved¹ with the original one.*

Proof. Consider the set D of edges that are delayed by this algorithm, and C the edges that are kept at their original position. Starting from the original sequence, we can delay all the edges of D by exactly ϵ . The flag filtration defined by this delayed sequence is obviously $(0, \epsilon)$ -interleaved with the original. We now run the regular backward algorithm on this sequence, with the difference that the edges in C are handled as if they were never dominated. The output filtration has the same persistence diagram as the delayed sequence, which is at distance at most ϵ from the diagram of the original filtration. The key observation here is that this output filtration is precisely what the approximation algorithm produces. ◀

7 Zigzag persistence

The filtrations we have discussed so far are increasing sequences of complexes. There exists a more general type of filtration, called zigzag filtration [8, 20] $\mathcal{Z} : K_1 \hookrightarrow K_2 \leftarrow K_3 \hookrightarrow \dots \hookrightarrow K_n$. Here consecutive complexes are still related by an inclusion, but the direction of this inclusion may be different for every consecutive pair. In other words, the complex K_i is obtained by either *inclusion* or *removal* of simplices from the previous complex K_{i-1} . Persistence diagrams can still be defined for these filtrations. Again, in this paper, we are only interested in flag zigzag filtrations, where each complex is a clique complex. For a flag zigzag filtration the underlying graphs are related through inclusion or removal of edges. We show that edge collapse can again be used for simplification of such sequences.

In the case of standard persistence (explained in Section 4) the goal of the simplification process was to shift as many dominated edges as possible towards the end of a filtration and then trim them. For a zigzag flag filtration there are several possible ways to simplify it: 1. If a dominated edge is included and is never removed, then as usual we try to *shift* it towards the end and trim it. 2. If an edge is included and removed both as dominated, then we try to *shift* the inclusion till its removal and then annihilate both operations. 3. If an edge is included as non-dominated but later removed as dominated then we try to shift its removal towards the right till the end or its re-insertion. 4. A zigzag filtration is symmetric and a removal is an inclusion from the opposite direction, therefore, we can shift dominated removals towards the beginning and perform symmetric operations as in 2.

The 3rd method reduces the number of events at the cost of a slightly bigger complex, which may or may not be preferred over a more “zigzaggy” filtration, so we do not use it in the default algorithm.

¹ See [9] for a definition of interleaving.

With more ways to simplify, the simplification process of a zigzag flag filtration is more delicate compared to the usual filtration. And it has some subtleties, first, can we shift a dominated edge inclusion across an edge removal? We show that (in Lemma 8), a dominated edge e can be shifted across an edge removal if e is also dominated after the edge removal. Resolving the first issue leads us to the question, how to index (order) inclusions and removals of the same grade? In practice, this situation is not common and two complexes at consecutive grades are linked through either inclusions or removals. Therefore, we adopt the following representation for a zigzag flag filtration.

We will use the same notations $t_i, G_{t_i}, \overline{G}_{t_i}$ and E as in Section 3. We represent a zigzag filtration in slightly more general way as $\mathcal{Z} : \overline{G}_{t_1} \leftarrow \overline{G}_{t'_1} \hookrightarrow \overline{G}_{t_2} \leftarrow \dots \hookrightarrow \overline{G}_{t_{i-1}} \leftarrow \overline{G}_{t'_{i-1}} \hookrightarrow \overline{G}_{t_i} \leftarrow \overline{G}_{t'_i} \hookrightarrow \overline{G}_{t_{i+1}}, \dots \hookrightarrow \overline{G}_{t_n}$. Here $G_{t'_i}$ is an intermediate graph at grade t_i . In a usual zigzag, $\overline{G}_{t'_i}$ is equal to either \overline{G}_{t_i} or $\overline{G}_{t_{i+1}}$ depending on the direction of the arrow. Note that the standard zigzag algorithm still applies to this version.

Below, we provide a sufficient condition to shift and swap an inclusion with removal.

► **Lemma 8** (Zigzag Shifting-Swapping Lemma). *Let e be an edge inserted at t_i , $e \in G_{t'_i}$ and dominated in both graphs G_{t_i} and $G_{t'_i}$. Then the persistence diagrams of the original zigzag flag filtration $\{ \dots \leftarrow \overline{G}_{t'_{i-1}} \xrightarrow{e} \overline{G}_{t_i} \leftarrow \overline{G}_{t'_i} \hookrightarrow \overline{G}_{t_{i+1}} \leftarrow \dots \}$ and the shifted-swapped sequence $\{ \dots \leftarrow \overline{G}_{t'_{i-1}} \hookrightarrow \overline{G}_{t_i} \setminus e \leftarrow \overline{G}_{t'_i} \setminus e \xrightarrow{e} \overline{G}_{t_{i+1}} \leftarrow \dots \}$ are equivalent. That is, the grade of e can be shifted to t_{i+1} .*

Proof. The proof follows through a similar argument as Lemma 2. All three squares in the following diagram commute as all the maps are induced by inclusions. Note that the top left and the bottom right horizontal maps can be induced by non-elementary inclusions.

$$\begin{array}{ccccccc}
 H_p(\overline{G}_{t'_{i-1}}) & \xleftarrow{|e|^*} & H_p(\overline{G}_{t_i}) & \xleftarrow{\quad} & H_p(\overline{G}_{t'_i}) & \xrightarrow{\quad} & H_p(\overline{G}_{t_{i+1}}) \\
 \parallel & & \begin{array}{c} |e|^* \uparrow \\ |rt|^* \downarrow \end{array} & & \begin{array}{c} |e|^* \uparrow \\ |rt|^* \downarrow \end{array} & & \parallel \\
 H_p(\overline{G}_{t'_{i-1}}) & \xrightarrow{\quad} & H_p(\overline{G}_{t_i} \setminus e) & \xleftarrow{\quad} & H_p(\overline{G}_{t'_i} \setminus e) & \xleftarrow{|e|^*} & H_p(\overline{G}_{t_{i+1}})
 \end{array}$$

Since the vertical maps are either equalities or isomorphisms induced by the inclusion of the dominated e ($|rt|$ is the corresponding retraction map associated with the collapse), the result follows immediately. That is, the shift of e to the grade t_{i+1} preserves the diagram. ◀

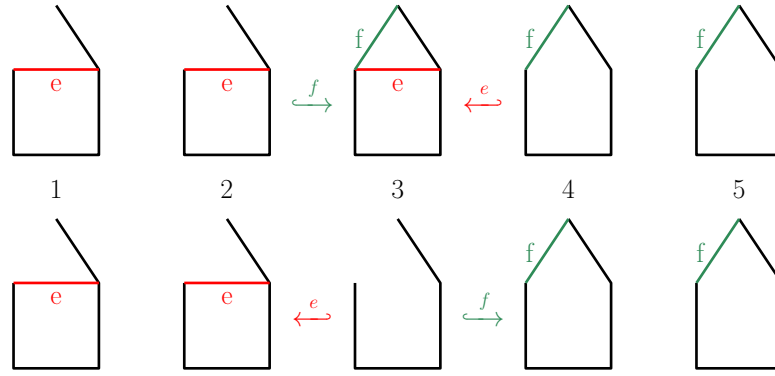
Note that in the above lemma, the hypothesis that edge e should be dominated in the graph $G_{t'_i}$ is necessary as shown in Figure 1.

Simultaneous insertion and removal of a dominated edge can be canceled.

► **Lemma 9** (Cancellation Lemma). *Let e be an edge inserted and removed at t_i . If e is dominated in G_{t_i} , then the persistence diagrams of the following two sequences $\{ \dots \leftarrow \overline{G}_{t'_{i-1}} \xrightarrow{e} \overline{G}_{t_i} \xleftarrow{e} \overline{G}_{t'_i} \hookrightarrow \overline{G}_{t_{i+1}} \leftarrow \dots \}$ and $\{ \dots \leftarrow \overline{G}_{t'_{i-1}} \hookrightarrow \overline{G}_{t_i} \setminus e \leftarrow \overline{G}_{t'_i} \hookrightarrow \overline{G}_{t_{i+1}} \leftarrow \dots \}$ are the same.*

Algorithm. Algorithm 2 to simplify $\mathcal{Z} : \overline{G}_{t_1} \leftarrow \dots \overline{G}_{t_i} \leftarrow \overline{G}_{t'_i} \hookrightarrow \overline{G}_{t_{i+1}}, \dots \hookrightarrow \overline{G}_{t_n}$ is again a combination of swapping, shifting, trimming and cancelling of a dominated edge. For each edge e in \mathcal{Z} there is a list of pairs $\langle t, inc \rangle$ associated with it, where t is a grade and inc is a Boolean variable to denote whether e is inserted or removed at t . Below, we provide the main steps of the zigzag simplification algorithm. The algorithm first processes all the edge inclusions in decreasing grade order from t_n to t_1 and tries to shift them towards the

44:10 Swap, Shift and Trim to Edge Collapse a Filtration



■ **Figure 1** In the top sequence, the green edge f is dominated at grade 3 and non-dominated at grade 4. Shifting and swapping the inclusion of f with the removal of the red edge e results in the bottom sequence. This results in two different one dimensional persistence diagrams of the associated flag complexes. For the top sequence it is $\{[1, 5]\}$ and for the bottom $\{[1, 2], [4, 5]\}$. Note that it is standard to use closed intervals in a zigzag persistence diagram.

end. After processing the first edge inclusion, it processes all the removals in increasing grade order from t_1 to t_n and tries to shift them towards the beginning. This process can be repeated several times until it converges. We use $t(e)$ to denote the current grade of the edge e being considered by the algorithm.

■ **Algorithm 2** Core zigzag flag filtration algorithm.

```

1: procedure CORE-ZIGZAG-FLAG-FILTRATION( $E$ )
2:   for all edge inclusions, backward (from  $t_n$  to  $t_1$ ) do
3:     if the current edge  $e$  is dominated in the graph  $G_{t(e)}$  then
4:       if  $t(e) == t_n$  then
5:         trim  $e$  (delete the element  $\langle t(e), inc \rangle$ ).
6:       else if  $G_{t(e)} \neq G_{t'(e)}$  then  $\triangleright$  the next step is a removal  $G_{t(e)} \leftrightarrow G_{t'(e)}$ .
7:         if  $e \notin G_{t'(e)}$  then
8:           delete the inclusion-removal pair of  $e$  at  $t(e)$ .
9:         else if  $e$  is dominated in  $G_{t'(e)}$ . then
10:          set  $t(e) = t(e) + 1$  and go-to step 3.  $\triangleright t(e) + 1$  denotes the next grade.
11:        end if
12:      else  $\triangleright$  the next step is an inclusion  $G_{t(e)} \hookrightarrow G_{t(e)+1}$ .
13:        set  $t(e) = t(e) + 1$  and go-to step 3.
14:      end if
15:    end if
16:  end for
17:  Move forward from  $t_1$  to  $t_n$  and process edge removals symmetric to steps 2-16.
18: end procedure

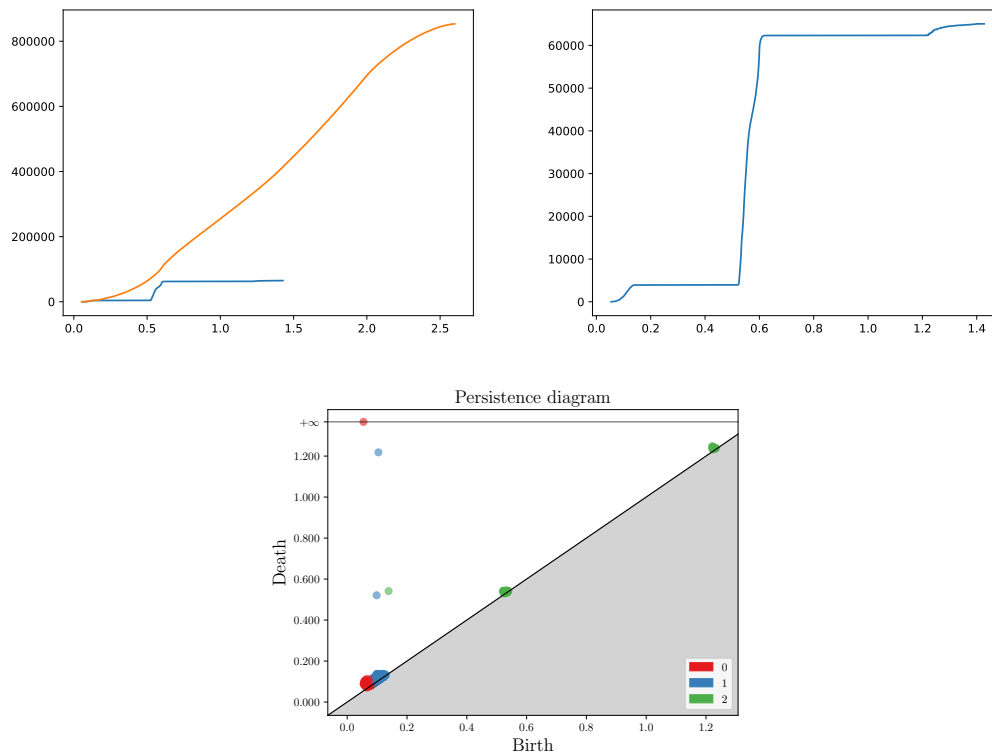
```

Note that an edge can be inserted and removed multiple times, in this case, the algorithm proceeds by pairing an inclusion with its next removal. Algorithm 2 outlines the essential aspects of the computation but is not optimal. Like Algorithm 1 we can use the Adjacency lemma (Lemma 4) to perform fewer domination checks. We can easily parallelize the zigzag simplification algorithm using the same divide and conquer approach described in Section 5.

8 Experiments

Complete graph. Starting from a complete graph on 700 vertices where all edges appear at the same time, the size of the graph after applying the algorithm several times decreases as 244650 (initial), 5340, 3086, 1307, 788 and finally 699. It stops decreasing after the 5th round since 699 edges is obviously minimal. This example demonstrates that one round of the algorithm is far from producing a fully reduced sequence. However, it removed a large number of edges, which makes subsequent rounds much faster, and may have already reduced the complex enough to compute (persistent) homology.

Torus: distribution of filtration values. We use a dataset with 1307 points on a torus embedded in \mathbb{R}^3 . Figure 2 (left) shows the distribution of the edge lengths. Originally, there are 853471 edges and the longest has size 2.6. We apply repeatedly the backward algorithm until the process converges. In the end, we are left with 65053 edges, and a maximal filtration value of 1.427.



■ **Figure 2** Filtration value of edges for a torus (top). Orange is for original edges and blue after collapse. Top right: enlarged blue graph. Bottom: persistence diagram.

First, note that some implementations (of which the first one is Eirene [18]) of Rips persistence first check at which filtration value the complex becomes a cone (here around 2) and ignore longer edges. In our algorithm, this check is performed implicitly and the long edges are dominated by the apex of the cone and thus get removed (we actually manage to go significantly lower than 2). Still, it remains sensible to avoid those edges when possible.

44:12 Swap, Shift and Trim to Edge Collapse a Filtration

After collapsing, we notice several regions in the curve. First some short edges are added progressively, until the complex gets the homotopy type of a torus. Then nothing happens for a while, until we have enough edges to kill one of the 1-cycles and fill the cavity, where many edges are inserted at the same time. Then again nothing happens while the complex is equivalent to a circle, until we can kill this last 1-cycle, and the process quickly stops with a contractible complex.

Benchmark backward vs forward. We benchmark the new backward algorithm with the forward algorithm. For the forward algorithm, we use the code from Giotto-ph [25], which is derived from our implementation in Gudhi but faster by a factor 1.5 to 2. Our benchmarking considers two aspects: run-time and reduction size (see Table 1). The datasets are: *uniform* for an i.i.d. sample of points in a square, *sparse* for the same, but using a low threshold on the maximal size of edges, *polygon* for a regular polygon, *circle* for an i.i.d. uniform sample of a circle, *dragon* comes from [24] and *O3* from [3] (the first version uses a threshold of 1.4 on edge lengths).

The backward algorithm comes with an optimization using a *dense* array indexed by vertices. This usually speeds things up nicely, but in cases where the original set of edges is very sparse, this dense array can be an issue, so we also have a version without this array, denoted *sparse*.

■ **Table 1** Run-time and reduction size comparison. Column *before* and *after* contains the number of edges before and after collapsing, and column *time* contains run time in seconds of the collapse.

	vertices	Forward			Backward		
		before	after	time	after	time dense	time sparse
uniform	1000	499500	2897	2.4	2897	1.7	2.4
sparse	50000	389488	125119	0.3	125119	1.9	0.17
polygon	300	44850	44701	3.6	44701	0.5	1
circle	300	44850	41959	4.8	41959	0.4	0.8
complete	900	404550	24540	43	5980	0.4	0.4
torus	1307	853471	94993	31	94993	3.2	5
dragon	2000	1999000	53522	29	53522	14	20
O3 (1.4)	4096	4107941	13674	59	13674	37	51
O3	1024	523776	519217	200	519217	12	23

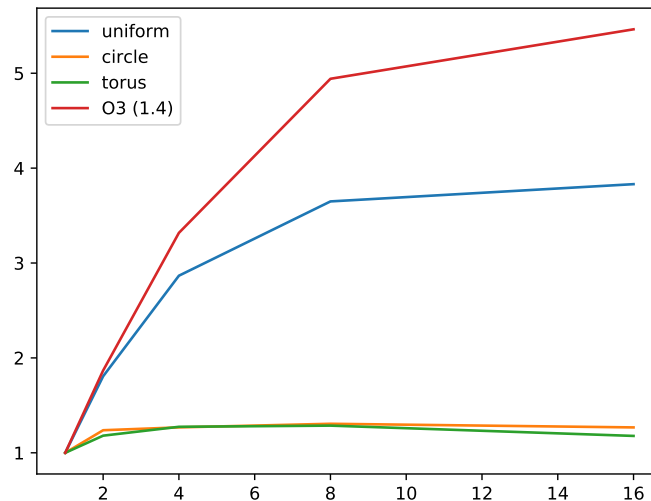
Table 1 shows a clear advantage for the backward algorithm in cases where few edges can be removed, or when several edges have the same filtration value. Except for *complete* which is a plain complete graph with every edge at the same filtration value, all edges are computed as Euclidean Rips graphs.

When all the input edges have distinct filtration values, both algorithms output exactly the same list of edges. However, this isn't the case anymore when multiple edges have the same filtration value (and in particular if we apply the algorithm several times). The forward algorithm, as presented, relies on the order of the edges and does not take advantage of edges with the same filtration value. The backward algorithm, at its core, checks if an edge is dominated *at a specific filtration value (grade)*. As seen in Table 1, for a complete graph on 900 vertices, the backward algorithm outputs 5 times fewer edges than the forward algorithm.

■ **Table 2** Gains with the approximate algorithm, for different interleaving factors.

	original	1 (exact)	1.01	1.1	1.5	2	10	100
uniform	499500	2897	2891	2859	2609	2462	2356	2353
circle (seconds)	44850	42007 <i>0.4</i>	30423 <i>0.33</i>	20617 <i>0.22</i>	17552 <i>0.16</i>	16404 <i>0.14</i>	14574 <i>0.12</i>	14342 <i>0.115</i>
dragon	1999000	53522	52738	52161	45439	40564	36094	35860
O3 (1.4)	4107941	13674	13635	13418	12682	12050	11828	11823

Size gains with approximate version. Table 2 shows the number of remaining edges when we don't require the output to have the same persistence diagram, but only ask that the modules be multiplicatively α -interleaved. Usually, the approximate version gives modest gains over the exact version, for roughly the same running time. However, in some cases that are hard to simplify like the circle, even a small error allows a significant number of collapses.



■ **Figure 3** Speed gain in function of the number of threads.

Parallelism benchmark. We wrote a limited² prototype based on `tbb::parallel_reduce` and tested it on an i7-10875H CPU (8 cores, 16 threads) by limiting the number of threads. Figure 3 shows promising results for some datasets, but also that there is room for better parallel algorithms.

Persistence benchmark. In our experience, doing edge collapses before computing persistent homology helps a lot for (homology) dimension 2 or higher. However, it is a terrible idea if we only care about dimension 0. The case of dimension 1 is more mixed, it can help in some cases and hurt in others. By default we would only recommend its use for dimension greater than or equal to 2.

² This implementation assumes that no two edges have the same filtration value.

For convenience, the persistence computation is done using the version of Ripser [3] found in giotto-ph [25] with $n_threads = 1$, and with our new backward algorithm. This means that edges after the complex has become a cone are ignored. Table 3 shows the time it takes to compute persistent homology in dimension up to k , either directly, or first collapsing before computing it.

■ **Table 3** Persistent homology computation time in seconds, with or without edge collapse.

	dim 1	collapse & dim 1	dim 2	collapse & dim 2	collapse & dim 3
torus3D	6.2	3.8	75	6.4	47
dragon	3.3	9.2	148	9.7	16.3

References

- 1 M. Aggarwal and V. Periwai. Dory: Overcoming barriers to computing persistent homology, 2021. [arXiv:2103.05608](https://arxiv.org/abs/2103.05608).
- 2 J. A. Barmak and E. G. Minian. Strong homotopy types, nerves and collapses. *Discrete and Computational Geometry*, 47:301–328, 2012. doi:10.1007/s00454-011-9357-5.
- 3 U. Bauer. Ripser: efficient computation of Vietoris-Rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021. doi:10.1007/s41468-021-00071-5.
- 4 U. Bauer, M. Kerber, J. Reininghaus, and H. Wagner. PHAT - persistent homology algorithms toolbox. *Journal of Symbolic Computation*, 78, 2017. doi:10.1016/j.jsc.2016.03.008.
- 5 J-D. Boissonnat and S. Pritam. Computing persistent homology of flag complexes via strong collapses. *International Symposium on Computational Geometry (SoCG)*, 2019. doi:10.4230/LIPIcs.SocG.2019.55.
- 6 J-D. Boissonnat and S. Pritam. Edge collapse and persistence of flag complexes. *International Symposium on Computational Geometry (SoCG)*, 2020. doi:10.4230/LIPIcs.SocG.2020.19.
- 7 M. Botnan and G. Spreemann. Approximating persistent homology in Euclidean space through collapses. In: *Applicable Algebra in Engineering, Communication and Computing*, 26:73–101, 2015. doi:10.1007/s00200-014-0247-y.
- 8 G. Carlsson and V. de Silva. Zigzag persistence. *Found Comput Math*, 10, 2010. doi:10.1007/s10208-010-9066-0.
- 9 F. Chazal, V. de Silva, M. Glisse, and S. Oudot. *The Structure and Stability of Persistence Modules*. SpringerBriefs in Mathematics. Springer, Cham, 2016. doi:10.1007/978-3-319-42545-0.
- 10 F. Chazal and S. Oudot. Towards persistence-based reconstruction in Euclidean spaces. *International Symposium on Computational Geometry (SoCG)*, 2008. doi:10.1145/1377676.1377719.
- 11 A. Choudhary, M. Kerber, and S. Raghvendra. Polynomial-sized topological approximations using the permutahedron. *Discrete and Computational Geometry*, 61:42–80, 2019. doi:10.1007/s00454-017-9951-2.
- 12 H. Derksen and J. Weyman. Quiver representations. *Notices of the American Mathematical Society*, 52(2):200–206, February 2005. URL: <https://www.ams.org/journals/notices/200502/fea-weyman.pdf>.
- 13 T. K. Dey, D. Shi, and Y. Wang. Simba: An efficient tool for approximating Rips-filtration persistence via simplicial batch collapse. *ACM J. Exp. Algorithmics*, 24, January 2019. doi:10.1145/3284360.
- 14 P. Dłotko and H. Wagner. Simplification of complexes for persistent homology computations. *Homology, Homotopy and Applications*, 16:49–63, 2014. doi:10.4310/HHA.2014.v16.n1.a3.
- 15 H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.
- 16 Gudhi: Geometry understanding in higher dimensions. URL: <https://gudhi.inria.fr/>.

- 17 A. Hatcher. *Algebraic Topology*. Univ. Press Cambridge, 2001. URL: <https://pi.math.cornell.edu/~hatcher/AT/ATpage.html>.
- 18 A. Hylton, G. Henselman-Petrusek, J. Sang, and R. Short. Tuning the performance of a computational persistent homology package. *Software: practice & experience*, 49(5):885–905, May 2019. doi:10.1002/spe.2678.
- 19 M. Kerber and R. Sharathkumar. Approximate Čech complex in low and high dimensions. In *Algorithms and Computation*, pages 666–676. By Leizhen Cai, Siu-Wing Cheng, and Tak-Wah Lam. Vol. 8283. Lecture Notes in Computer Science, 2013. doi:10.1007/978-3-642-45030-3_62.
- 20 C. Maria and S. Oudot. Zigzag persistence via reflections and transpositions. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 181–199, January 2015. doi:10.1145/1542362.1542408.
- 21 K. Mischaikow and V. Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete and Computational Geometry*, 50:330–353, September 2013. doi:10.1007/s00454-013-9529-6.
- 22 D. Mozozov. Dionysus. URL: <http://www.mrzv.org/software/dionysus/>.
- 23 J. Munkres. *Elements of Algebraic Topology*. Perseus Publishing, 1984.
- 24 N. Otter, M. Porter, U. Tillmann, P. Grindrod, and H. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science, Springer Nature*, 6:17, 2017. doi:10.1140/epjds/s13688-017-0109-5.
- 25 J. B. Pérez, S. Hauke, U. Lupo, M. Caorsi, and A. Dassatti. Giotto-ph: A Python Library for High-Performance Computation of Persistent Homology of Vietoris-Rips Filtrations. *CoRR*, 2021. arXiv:2107.05412.
- 26 M. Xiao S. Zhang and H. Wang. GPU-Accelerated Computation of Vietoris-Rips Persistence Barcodes. *International Symposium on Computational Geometry (SoCG)*, 2020. doi:10.4230/LIPIcs.SocG.2020.70.
- 27 D. Sheehy. Linear-size approximations to the Vietoris–Rips filtration. *Discrete and Computational Geometry*, 49:778–796, 2013. doi:10.1007/s00454-013-9513-1.
- 28 A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33:249–274, 2005. doi:10.1007/s00454-004-1146-y.

Hardness and Approximation of Minimum Convex Partition

Nicolas Grelier ✉

Department of Computer Science, ETH Zürich, Switzerland

Abstract

We consider the Minimum Convex Partition problem: Given a set P of n points in the plane, draw a plane graph G on P , with positive minimum degree, such that G partitions the convex hull of P into a minimum number of convex faces. We show that Minimum Convex Partition is NP-hard, and we give several approximation algorithms, from an $\mathcal{O}(\log OPT)$ -approximation running in $\mathcal{O}(n^8)$ -time, where OPT denotes the minimum number of convex faces needed, to an $\mathcal{O}(\sqrt{n} \log n)$ -approximation algorithm running in $\mathcal{O}(n^2)$ -time. We say that a point set is k -directed if the (straight) lines containing at least three points have up to k directions. We present an $\mathcal{O}(k)$ -approximation algorithm running in $n^{\mathcal{O}(k)}$ -time. Those hardness and approximation results also holds for the Minimum Convex Tiling problem, defined similarly but allowing the use of Steiner points. The approximation results are obtained by relating the problem to the Covering Points with Non-Crossing Segments problem. We show that this problem is NP-hard, and present an FPT algorithm. This allows us to obtain a constant-approximation FPT algorithm for the Minimum Convex Partition Problem where the parameter is the number of faces.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases degenerate point sets, point cover, non-crossing segments, approximation algorithm, complexity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.45

Related Version *Full Version*: <https://arxiv.org/abs/1911.07697>

Funding Research supported by the Swiss National Science Foundation within the collaborative DACH project *Arrangements and Drawings* as SNSF Project 200021E-171681.

Acknowledgements The author thanks Michael Hoffmann for his helpful advice.

1 Introduction

The CG Challenge 2020 organised by Demaine, Fekete, Keldenich, Krupke and Mitchell [5], was about solving instances of *Minimum Convex Partition* (MCP).

► **Definition 1** (Demaine et al. [5]: Minimum Convex Partition problem). Given a set P of n points in the plane. The objective is to compute a plane graph with vertex set P (with each point in P having positive degree) that partitions the convex hull of P into the smallest possible number of convex faces. Note that collinear points are allowed on face boundaries, so all internal angles of a face are at most π .

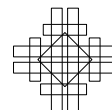
As explained by Bose et al., this problem has applications in routing [3]. They show that a routing algorithm named *Random-Compass* that works for triangulations can be extended to convex partitions. Having a convex partition with few faces reduces the amount of data to store. From now on, we denote by P a set of n points in the plane.

In this paper, we present several approximation algorithms for MCP. We obtain those approximation algorithms by relating the MCP problem to the *Covering Points with Non-Crossing Segments* (CPNCS) problem. First, we define what *non-crossing segments* are.



© Nicolas Grelier;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 45; pp. 45:1–45:15
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



► **Definition 2** (Non-Crossing Segments). We call a part of a (straight) line bounded by two points a *segment*. The two points are referred to as *endpoints* of the segment. Note that we do not force the endpoints to be distinct, therefore we consider a point p as being a segment. The endpoint of p is p itself. Two segments are *non-crossing* if the intersection of their relative interior is empty.

► **Definition 3** (Covering Points with Non-Crossing Segments). Given a set P of n points, find a minimum number of non-crossing segments whose endpoints are in P such that each point of P is contained in at least one segment.

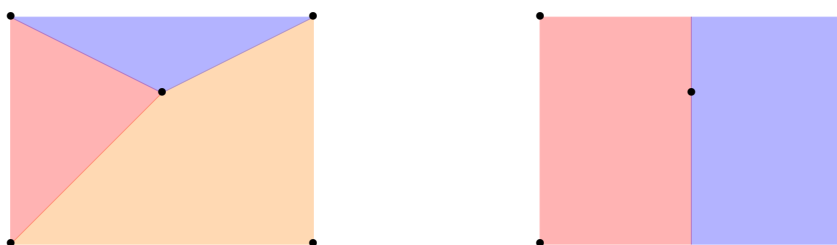
The condition that the endpoints of the segments must be in P has no effect on the number of segments required. We add it as it simplifies some arguments. Note that CPNCS is not a so-called *set cover problem* nor an *exact cover problem*. We believe that CPNCS is interesting in itself. Even though it is a very natural problem, to the best of our knowledge it had not been introduced before.

1.1 NP-hardness results

Fevens, Meijer and Rappaport first considered the MCP problem in 2001 [7], and its complexity was explicitly asked about by Knauer and Spillner in 2006 [12]. It has remained open since then [2, 5]. We show in Section 5 that MCP is NP-hard. To do this, we use the decision version of the problem, as stated below:

► **Definition 4** (MCP - decision version). Given a set P of points in the plane and a natural number k , is it possible to find at most k closed convex polygons whose vertices are points of P , with the following properties: *a)* The union of the polygons is the convex hull of P , *b)* the interiors of the polygons are pairwise disjoint, and *c)* no polygon contains a point of P in its interior.

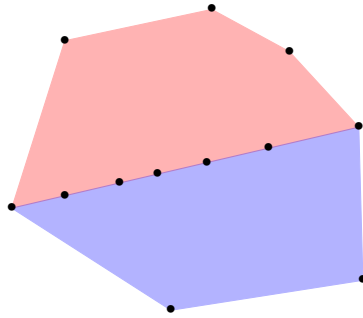
We also show NP-hardness of a similar problem, which we call *Minimum Convex Tiling* problem (MCT). The problem is exactly as in Definition 4, but the constraint about the vertices of the polygons is removed (i.e. they need not be points of P). This can make a difference as shown in Figure 1. Equivalently, the MCT problem corresponds to the MCP problem when Steiner points are allowed. A *Steiner point* is a point that does not belong to the point set given as input, and which can be used as a vertex of some polygons. The MCT problem has been studied in 2012 by Dumitrescu, Har-Peled and Tóth, who asked about the complexity of the problem [6]. We answer their question, and our proofs are very similar for MCP and MCT. We also show in the full version of the paper [10] that CPNCS is NP-hard, even for some constrained point sets.



■ **Figure 1** A minimum partition with three convex polygons and a tiling with two.

1.2 Approximation algorithms

For the related problem *Minimum Convex Partition of Polygons with Holes*, Bandyapadhyay, Bhowmick and Varadarajan showed the existence of a $(1 + \varepsilon)$ -approximation algorithm running in time $n^{\mathcal{O}((\log n/\varepsilon)^4)}$ [1]. Although they only consider holes with non empty interior, one can observe that their proof extends to the case of point holes. This is an even more general setting than MCP for point sets, so their algorithm also applies in our setting. This implies that MCP is not APX-hard unless $NP \subseteq DTIME(2^{\text{polylog } n})$.



■ **Figure 2** The number of inner points can be arbitrarily much larger than the number of convex faces required.

Under the assumption that no three points are collinear, Knauer and Spillner have shown the existence of a $\frac{30}{11}$ -approximation algorithm for MCP in 2006 [12]. As a lower bound on the number of convex faces for one particular point set, they rely on the observation that each inner point has degree at least 3. The *inner points* of P are the points not on the boundary of the convex hull. This gives a lower bound on the number of edges, and therefore on the number of faces, by Euler's formula. Note that the restriction that no three points are on a line is necessary, as shown in Figure 2. There are only two faces in a minimum convex partition of this point set, and all the inner points have degree 2.

Additionally, Knauer and Spillner showed how to adapt any constructive upper bound on the number of faces into an approximation algorithm. More explicitly, they showed that if one can compute in polynomial time a convex partition with at most λn convex faces, then there exists a 2λ -approximation algorithm running in polynomial time. The best result to date is a proof by Sakai and Urrutia that one can partition a point set in quadratic time using at most $\frac{4}{3}n$ convex faces (the result was presented at the 7th JCCGG in 2009, the paper appeared on arXiv in 2019) [19]. Although they do not mention it, combining this result with the one by Knauer and Spillner gives a quadratic time $\frac{8}{3}$ -approximation algorithm.

Concerning previous upper bounds, Neumann-Lara, Rivero-Campo and Urrutia first showed in 2004 how to construct in quadratic time a partition of any point set with at most $\frac{10}{7}n$ convex faces [17]. In 2006, Knauer and Spillner improved this to $\frac{15}{11}n$ convex faces [12]. As said above, the best known upper bound is $\frac{4}{3}n$, as proven by Sakai and Urrutia in 2009.

Relatedly for lower bounds, García-Lopez and Nicolás have given in 2013 a construction of point sets for which any convex partition has at least $\frac{35}{32}n - \frac{3}{2}$ faces [8].

All these results concerning upper bounds hold for all point sets, even where many points are on a line. Indeed, slightly shifting the points so that no three points are on a line can only increase the number of convex faces needed. So an upper bound for point sets where no three points are on a line also holds for all point sets. However, as mentioned above, the lower bound used by Knauer and Spillner does not extend to our setting, where we consider all point

sets. They say that a constant-approximation algorithm would be desirable for unrestricted point sets, but so far not even an $\mathcal{O}(n^{1-\varepsilon})$ -approximation is known. For the MCT problem, Dumitrescu, Har-Peled and Tóth showed the existence of a 3-approximation algorithm for point sets with no three collinear points [6]. They also ask whether a constant-approximation algorithm exists when this constraint is removed. However, so far no $\mathcal{O}(n^{1-\varepsilon})$ -approximation algorithm is known. In Section 3, we prove the following:

► **Theorem 5.** *There exists $\mathcal{O}(\log OPT)$ -approximation algorithms for MCP, MCT and CPNCS running in $\mathcal{O}(n^8)$ -time.*

Allowing several points to be on a line does not simply create tedious technicalities to deal with. The crux of the matter is to find, for a fixed point set, an exploitable lower bound on the number of faces in a minimum convex partition. When no three points are on a line, the number of inner points in P gives a linear lower bound on the number of faces in a convex partition [12], and in a convex tiling [6]. In this paper, we consider point sets with no restriction. We introduce the CPNCS problem as it pinpoints where the difficulty of finding a constant-approximation algorithm for MCP is and makes the problem easier to study. We show in Section 2 the following theorem, which is used to prove Theorem 5:

► **Theorem 6.** *Let P be a set of n points with at least one inner point, and let $\lambda \geq 1$ be a real number. Let f_m denote the minimum number of faces in a convex partition of P . Let s_m denote the minimum number of non-crossing segments in a covering of the inner points of P , denoted by P_i .*

1. *It holds that $\frac{s_m}{6} \leq f_m \leq 8s_m$.*
2. *Given a covering of P_i with $s \leq \lambda s_m$ non-crossing segments, it is possible to compute in $\mathcal{O}(n^2)$ -time a convex partition of P with at most $24\lambda f_m$ convex faces.*
3. *Given a convex partition of P with $f \leq \lambda f_m$ convex faces, it is possible to compute in $\mathcal{O}(n)$ -time a covering of P_i with at most $44\lambda s_m$ non-crossing segments.*

The theorem also holds when considering convex tilings instead of convex partitions.

The idea behind the similarity of MCP, MCT and CPNCS is that they are all about maximizing the number of vertices of degree 2 with incident edges being aligned in a plane straight-line drawing of a graph on a point set. We show in the full version of the paper [10] that MCP and CPNCS are however not equivalent, meaning that one cannot use an optimal solution for one to deduce an optimal solution for the other.

1.3 Exact algorithms, FPT algorithms

Under the assumptions that the points lie on the boundaries of a fixed number h of nested convex hulls, and that no three points lie on a line, Fevens, Meijer and Rappaport gave an algorithm for solving MCP in time $\mathcal{O}(n^{3h+3})$ [7]. Observe that this is not an FPT algorithm. Some integer linear programming formulations of the problem have been recently introduced [2, 20, 4].

A first FPT algorithm with respect to the number k of inner points was introduced by Grantson and Levcopoulos, with running time $\mathcal{O}(2^{16k} k^{6k-5} n)$ [9]. The idea of the algorithm is to enumerate all plane graphs on the inner points, and then for each to them to guess how to connect the inner points to points on the boundary of the convex hull. Another FPT algorithm with respect to the number of inner points was later found by Spillner, with running time $\mathcal{O}(2^k k^4 n^3 + n \log n)$ [21].

We show in Section 4 the existence of an FPT algorithm that checks whether there is a solution for CPNCS with at most k non-crossing segments, running in time $\mathcal{O}(2^{k^2} k^{7k} + n^4 \log n)$. By Theorem 6, this gives us a constant-approximation FPT algorithm for MCP

and MCT, where the parameter is the number of convex faces needed. Under the assumption that no three points are on a line, the number of faces in a minimum convex partition or in a minimum convex tiling is the same as the number of inner points, up to a constant multiplicative factor [12, 6]. However, without this assumption the number of inner points can be arbitrarily much larger than the minimum number of convex faces, as shown in Figure 2.

2 The relation between MCP, MCT and CPNCS

Throughout this section, we denote by P a point set in the plane. We denote by P_i the set of inner points of P . Let p be in P . If P and $P \setminus \{p\}$ do not have the same convex hull, we say that p is an *extreme point*. We denote by $P'_i \subseteq P_i$ the extreme points in P_i , where P_i denotes the inner points in P . Note that a point might lie on the boundary of the convex hull of a point set without being an extreme point. We say that P is *special* if $|P'_i| \leq 2$. Recall that for a given covering of a point set Q with non-crossing segments, we always assume that the endpoints of the segments are in Q .

► **Lemma 7.** *Let P be a set of n points that is not special. Given a covering K of P_i with s non-crossing segments, one can compute in $\mathcal{O}(n^2)$ -time a convex partition Σ of P with at most $4s + 2|P'_i|$ faces. Moreover every segment in K is the union of some edges in Σ .*

Due to space constraints, we postpone the proof of Lemma 7 to the full version of the paper [10]. The idea of the proof is to compute a constrained triangulation with respect to the segments of the covering. This gives us a convex partition of the inner points, and it remains to connect the points in P'_i to points on the boundary of the convex hull.

► **Lemma 8.** *Let P be a set of n points. Given a convex tiling Σ of P with f faces, one can compute in $\mathcal{O}(n)$ -time a covering K of P_i with at most $6f - 2|P'_i|$ non-crossing segments. Moreover every segment in K is the union of some edges in Σ .*

Proof. The proof is illustrated in Figure 3. Let us denote by $G_0 = (V_0, E_0)$ the plane graph corresponding to the convex tiling, where a point in V_0 is extreme or has degree at least 3. Observe that some points in V_0 might not be in P . Also, the relative interior of an edge in E_0 might overlap with points in P . We assume that G_0 is given with a doubly connected edge list (DCEL) structure. If there is an edge between two points on the boundary of the convex hull of V_0 , but not consecutive, we remove this edge. Note that this decreases the number of faces by 1, and does not break the convexity property. We denote by m the number of such edges that we have removed. We also remove from P all points contained in the relative interior of an edge between two points on the boundary of the convex hull. We denote by P''_i the extreme points in P_i that we have not removed. As an edge contains at most two points in P'_i , we have $|P''_i| \geq |P'_i| - 2m$. Using the DCEL structure, this can be done in $\mathcal{O}(n)$ -time. We have obtained a new graph $G = (V, E)$, and there are $f - m$ convex faces in G . We denote by Q the set of inner points that are of degree at least 3 in G . We set $k := |Q|$. Now observe that for each point p in P''_i , there exists at least one edge e in E with one endpoint in Q , one endpoint on the boundary of the convex hull, such that e overlaps with a point in P''_i . This is because if we consider p and the two lines going through p and one of the two consecutive vertices in P''_i (the one before p and the one after p when going around P''_i in clockwise order), they define a wedge that one edge must intersect because of convexity. The point in P''_i can be an endpoint of e or in its relative interior. If for a point $p \in P''_i$ there are several edges that satisfy the conditions, we choose one

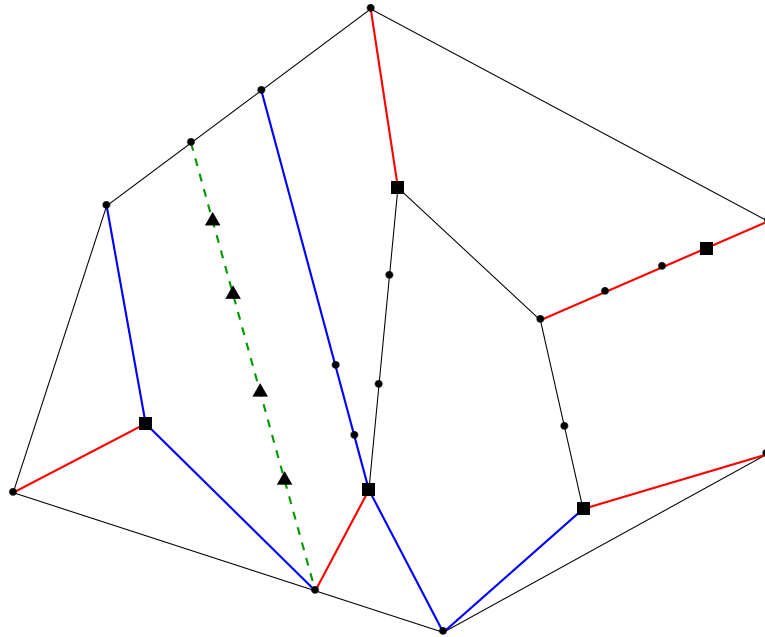


Figure 3 Illustration of Lemma 8. The green dashed edge and the triangle points are removed at the beginning for the analysis, and added back at the end. The extreme points in P_i'' are represented as square points. The edges in E' are in red. The other edges from P_i'' to the boundary of the convex hull are in blue.

arbitrarily. We denote these edges by E' . An edge in E' overlaps with exactly one point in P_i'' , thus $|E'| = |P_i''|$. We denote by E_b the edges not in E' that have a point on the boundary of the convex hull and the other in Q , and we denote $|E_b|$ by m' . The vertices on the boundary of the convex hull are adjacent to two other vertices on the boundary of the convex hull. Moreover, those vertices are incident to $|P_i''| + m'$ additional edges. We have $2|E| = \sum_{v \in V} \deg(v) \geq 3k + 2(n - k) + |P_i''| + m' = k + 2n + |P_i''| + m'$. By Euler's formula, we have $f - m = |E| - n + 1 \geq \frac{k + |P_i''| + m'}{2} + 1$.

Now, the solution consists of the union of all edges in E incident to two points in Q , with the m edges in E_0 that we have removed, and with the $|P_i''| + m'$ edges in $E' \cup E_b$. We may need those edges as they might overlap with points in P_i . Note that there are at most $3k$ edges in E incident to two points in Q as G is plane. Moreover, all points in P_i are indeed covered by the edges in our solution. Thus, we obtain a covering of P_i with s segments, where $s \leq 3k + m + m' + |P_i''| \leq 3(2(f - m) - |P_i''| - m') + m + m' + |P_i''| \leq 6f - 5m - 2|P_i''| \leq 6f - 5m - 2(|P_i''| - 2m) \leq 6f - 2|P_i''|$. ◀

It is now possible to combine Lemmas 7 and 8 to prove Theorem 6. The proof can be found in the full version of the paper [10].

3 Approximation algorithms for CPNCS

We present several approximation algorithms for CPNCS. Let us first consider the ones whose approximation ratio is not output-dependent. The best algorithms in terms of approximation ratio are constant-approximation algorithms. The fastest algorithms take quadratic time. Therefore by 2. of Theorem 6, all the algorithms we present for CPNCS can be used to

obtain approximation algorithms for MCP and MCT with the same order of approximation ratio, and the same order of running time. We have also one algorithm for CPNCS which realises an $\mathcal{O}(\log OPT)$ -approximation in time $\mathcal{O}(n^8)$, where OPT denotes the minimum number of segments needed. Using 1. and 2. of Theorem 6, we also derive from it the $\mathcal{O}(\log OPT)$ -approximation algorithm for MCP and MCT running in time $\mathcal{O}(n^8)$, where now OPT denotes the minimum number of faces needed in a convex partition, or in a convex tiling, respectively. This is how we prove Theorem 5. We first mention an easy approximation algorithm running relatively fast, at the cost of a high approximation ratio. The proof can be found in the full version of the paper [10]. The idea is to use the greedy algorithm to solve *Covering Points with lines* on P (a set cover problem), and then to split the lines into non-crossing segments.

► **Theorem 9.** *There exists an $\sqrt{n} \log(n)$ -approximation algorithm for CPNCS running in $\mathcal{O}(n^2)$ -time.*

Mitchell presented in a technical report some approximation algorithms for the problem of covering a point set with a minimum number of pairwise-disjoint triangles [16]. In his problem, the triangles of the covering must be subtriangles of some triangles given as input, for otherwise the problem would be trivial. He makes the assumption that no three points are on a line. We adapt his algorithms to our setting of CPNCS for point sets with no constraint. Let P be a set of n points. By doing a rotation if necessary, we can assume that no two points in P have the same x -coordinate. We say that a trapezoid is *constrained* if 1) it has two disjoint vertical sides, each lying on a line that contains a point in P , and 2) the two remaining sides are lying on lines that contain each at least two points in P . Note that there are $\mathcal{O}(n^6)$ constrained trapezoids.

We also allow for some degeneracies. Let us consider a triangle with vertices a , b and c , not all three on a line. If a is in P , the segment with endpoints b, c is vertical and lies on a line that contains a point in P , and the segments with endpoints a, b and a, c respectively are contained in some lines ℓ and ℓ' such that ℓ and ℓ' contains at least two points in P , then we say that the triangle is a constrained trapezoid. If a constrained trapezoid is split into two halves by a vertical line ℓ going through its interior, with ℓ containing a point in P , we obtain two constrained trapezoids. Likewise, if a segment s is in a constrained trapezoid τ , such that s lies on a line that contains at least two points in P , s intersects the interior of τ , and the endpoints of s are contained in the vertical sides of τ , then s splits τ into two constrained trapezoids.

For a set of points P where no two points have the same x -coordinate, we define the *enclosing trapezoid* as follows. Let ℓ_1 be the vertical line that contains the leftmost point in P , and let ℓ_2 be the vertical line that contains the rightmost point in P . Let L be the set of all lines containing at least two points in P . Observe that no line in L is vertical. We denote by a the highest intersection point between ℓ_1 and a line in L . We denote by b the lowest point intersection point between ℓ_1 and a line in L . Similarly, we denote by c and d , respectively, the highest intersection point, respectively the lowest intersection point, between ℓ_2 and a line in L . We denote by ℓ_3 the line containing a and c , and by ℓ_4 the line containing b and d . The *enclosing trapezoid* of P is the constrained trapezoid of $P \cup \{a, b, c, d\}$ defined by ℓ_1 , ℓ_2 , ℓ_3 and ℓ_4 . It is denoted by \mathcal{T}_P .

We define the *strong guillotine property* in the special case of segments. We show that if there is a covering of P with s non-crossing segments, then there is a covering of S with $\mathcal{O}(s \log s)$ non-crossing segments having the strong guillotine property. We then present an algorithm that outputs an optimal solution among all the coverings with non-crossing segments having the strong guillotine property. Let S be a set of non-crossing segments

covering P . We assume that the endpoints of the segments in S are in P . We say that S has the strong guillotine property with respect to a constrained trapezoid \mathcal{T} that contains all segments in S if a) S contains at most one segment, or if b) there exists a partitioning line ℓ containing at least two points in P and at least one segment in S , such that for any segment $s \in S$, ℓ either contains s or does not intersect the relative interior of s , and ℓ splits \mathcal{T} into two constrained trapezoids \mathcal{T}_1 and \mathcal{T}_2 , such that the segments in \mathcal{T}_1 , respectively \mathcal{T}_2 , have the strong guillotine property with respect to \mathcal{T}_1 , respectively \mathcal{T}_2 , or if c) there exists a vertical line not intersecting with the relative interior of any segment in S , that splits \mathcal{T} into two constrained trapezoids \mathcal{T}_1 and \mathcal{T}_2 , such that the segments in \mathcal{T}_1 , respectively \mathcal{T}_2 , have the strong guillotine property with respect to \mathcal{T}_1 , respectively \mathcal{T}_2 . Observe that the line ℓ in case b) only intersects the vertical sides of \mathcal{T} , for otherwise ℓ would not split \mathcal{T} into constrained trapezoids. We simply say that S has the strong guillotine property if it has the strong guillotine property with respect to the enclosing trapezoid \mathcal{T}_P .

► **Lemma 10.** *If there exists a covering of P with s non-crossing segments, then there exists a covering of P with $\mathcal{O}(s \log(s))$ non-crossing segments with the strong guillotine property.*

Proof. Recall that we assume that the endpoints of the segments are in P , by cropping them if need be. We can even crop some segments further such that they are pairwise-disjoint (it may be that now some segments are reduced to points). Consider the endpoints of the segments in that covering, that we denote by P' . We denote $|P'|$ by n' , and we have $n' \leq 2s$. Note that no two points in P' have the same x -coordinate. We denote by X the set of x -coordinates of the points in P' . We now consider the segment tree based on X , as defined in [18]. The segment tree defines some canonical intervals. Each interval, whose endpoints are in X , is partitioned into $\mathcal{O}(\log s)$ canonical intervals. We partition each segment in the covering, such that the projection on the x -axis of each new segment is a canonical interval. Therefore we obtain a covering of P with $\mathcal{O}(s \log(s))$ non-crossing segments. We claim that this family of segments has the strong guillotine property. Let us denote by x_i , $1 \leq i \leq n'$ the elements in X , ordered by increasing value. We distinguish two cases. If there exists a segment σ whose projection on the x -axis is equal to the interval $[x_1, x_{n'}]$, then we recurse on the parts above and below σ which contain some segments. Observe that if $n' = 2$ we are done. If there is no such segment, then by definition of a segment tree, there is no segment in the covering whose relative interior intersects the vertical line ℓ with x -coordinate equal to $x_{\lfloor (1+n')/2 \rfloor}$. Thus we can recurse on the left and right side of ℓ . ◀

► **Theorem 11.** *There exists an $\mathcal{O}(\log(OPT))$ -approximation algorithm running in $\mathcal{O}(n^8)$ -time for CPNCS.*

Proof. We explain how to recursively compute a minimum covering of P with non-crossing segments under the constraint that the solution has the strong guillotine property. The approximation ratio for the CPNCS problem when this additional constraint is removed follows from Lemma 10. If P is empty, we return no segment, which is a valid solution. If P can be covered with a single segment, we return that segment. This can be tested in $\mathcal{O}(n^2)$ time using duality. Now let us assume that not all points in P are on a line. We compute the enclosing trapezoid \mathcal{T}_P of P . We consider the four vertices a, b, c, d of \mathcal{T}_P . We start by adding the segment with endpoints a, c , and the segment with endpoints b, d . Now all the points to cover are within the enclosing trapezoid \mathcal{T}_P . We distinguish two cases, according to whether a segment with endpoints on the vertical sides of \mathcal{T}_P is in a minimum covering with non-crossing segments having the strong guillotine property. If it is, we can add it to the solution and recurse on the two new constrained trapezoids. If no such segment is part of a

minimum solution, then there exists a vertical line ℓ that splits a minimum solution into two parts, such that ℓ does not intersect the relative interior of any segment in that minimum solution. We can recurse on the $\mathcal{O}(n)$ choices of splitting vertically the constrained trapezoid into two constrained trapezoids. For each of the $\mathcal{O}(n^2)$ recursions, we compute the number of segments corresponding to that solution, and we output the solution corresponding to the one that minimises the number of segments.

To optimise we can do dynamic programming, and solve first the thinnest constrained trapezoids (in terms of width on the x -axis). There are $\mathcal{O}(n^6)$ constrained trapezoids, and we take quadratic time for each of them, so the total running time is $\mathcal{O}(n^8)$. ◀

We prove the following theorem in the full version of the paper [10].

► **Theorem 12.** *There exists an $\mathcal{O}(\log(n))$ -approximation algorithm running in $\mathcal{O}(n^7)$ -time for CPNCS.*

We say that a point set P is k -directed if there exists a set D of k directions, such that for any line ℓ that contains at least three points in P , the direction of ℓ is in D . For convenience, for any set of directions D and any segment s reduced to a point, we say that the direction of s is in D . We say that a set of segments S has the *autopartition property* if $|S| \leq 1$, or if there exists a line ℓ which contains at least one segment in S , and splits S into two sets that have the autopartition property. The relative interior of a segment in S is either contained in ℓ or does not intersect ℓ . Tóth has shown that any set of s' disjoint segments having up to k directions have an autopartition of size $\mathcal{O}(s'k)$ [22]. Using this result and similar techniques to the ones of Theorem 11, we show in the full version of the paper [10] the following:

► **Theorem 13.** *There exists an $\mathcal{O}(k)$ -approximation algorithm for CPNCS in k -directed sets running in $n^{\mathcal{O}(k)}$. Furthermore, there exists a 4-approximation algorithm for CPNCS in 2-directed sets running in time $\mathcal{O}(n^5)$.*

4 Fixed-parameter algorithm for CPNCS

As mentioned in the introduction, there are known fixed-parameter algorithms for MCP, where the parameter is the number of inner points. We present here a fixed-parameter constant-approximation algorithm for MCP and MCT, where the parameter is the number of faces in a minimum convex partition or a minimum convex tiling, respectively. For point sets where no three points are on a line, the minimum number of convex faces is at least half the number of inner points [12], and the number of convex tiles is at list a sixth of the number of inner points [6]. However, as shown in Figure 2, when we allow for several points to be on a line, the number of inner points can be arbitrarily much larger than the number of convex faces in a minimum convex partition. If the number of inner points is significantly larger than the number of convex faces needed, our algorithm has a lower running time. We first show that CPNCS is in FPT.

► **Theorem 14.** *We can compute in time $\mathcal{O}(2^{k^2} k^{7k} + n^4 \log n)$ whether a point set P can be covered with at most k non-crossing segments, and to output such a covering if it exists.*

The proof uses a kernelisation technique presented by Langerman and Morin for *Covering Points with Lines* [13]. Assume there is a line ℓ that contains at least $k + 1$ points in P . Then in any covering of P with at most k lines, ℓ must be in the covering. Otherwise, we would need at least $k + 1$ lines to cover the points contained in ℓ . Now one can compute all of these lines that contain at least $k + 1$ points, dismiss all of the covered points, until no line

covers more than k of the remaining points. If there remains more than k^2 points, then there is no covering of the point set with at most k lines. Otherwise, one can compute every way of covering the $\mathcal{O}(k^2)$ remaining points, and check whether there is one that uses in total at most k lines. In our setting, we are looking for a covering with non-crossing segments, which makes it more difficult. Indeed, if a line ℓ contains at least $k + 1$ points, we only know that ℓ must contain at least one segment of the covering. This means that we cannot simply dismiss the points covered by such a line. Also, we have to be careful about crossings. To prove Theorem 14, we need several lemmas. For a point set P , we say that a segment s is a *P-segment* if its endpoints are in P . Recall that we only consider coverings of a point set P with non-crossing *P*-segments.

► **Definition 15.** Let P be a point set, and let s and t be two crossing *P*-segments. We denote by p the intersection of s and t . We determine four points in P , that we call the *points enclosing* p . There are two points on $s \cap P$ and two points on $t \cap P$. The two points on $s \cap P$, denoted by u and v , are such that the segment with endpoints u and v , which we denote by uv , is the shortest *P*-segment contained in s whose relative interior contains p . Likewise, the two points u' and v' are such that $u'v'$ is the shortest *P*-segment contained in t whose relative interior contains p . The points u, v, u' and v' are the points enclosing p .

► **Lemma 16.** *Given a set P of n points, it is possible to compute in time $\mathcal{O}(n^4 \log n)$ the pairs of crossing *P*-segments, to find whether their intersection p is in P , and to store the points enclosing p . Additionally, we can also store for each *P*-segment how many points in P they contain, and the list of those points.*

The proof of Lemma 16 can be found in the full version of the paper [10].

► **Lemma 17.** *Given a set P of n points, and a natural number k , it is possible to find in time $\mathcal{O}(2^{k^2} + n^4 \log n)$ either a certificate that there is no covering of P with at most k non-crossing segments, or to output a family \mathcal{F} of $\mathcal{O}(2^{k^2})$ sets S containing at most k non-crossing *P*-segments, with the following properties: For any fixed covering of P with at most k non-crossing *P*-segments, there exists a set S in \mathcal{F} such that a) any segment $s \in S$ contains at least $k + 1$ points in P , b) for each segment t of the covering, if $|P \cap t \cap s| \geq 2$ for some $s \in S$, then t is contained in s , and c) if a segment of the covering contains at least $k + 1$ points in P , then it is contained in a segment in S .*

Let P be a point set and let k be a natural number. Observe that if a set S of segments satisfies property a), then in a covering with at most k segments of P , each segment s in S contains at least one segment t of the covering, such that $|P \cap t| \geq 2$. Indeed if there exists a segment $s \in S$ such that for any segment t in the covering, we have that $s \cap t$ contains at most one point in P , then at least $k + 1$ segments are needed to cover the points in $P \cap s$. This implies that if S consists of m segments and satisfies properties a) and b), then there are at least m segments in the considered covering of P with non-crossing segments.

Proof of Lemma 17. We first do some preprocessing by using the algorithm of Lemma 16. This takes $\mathcal{O}(n^4 \log n)$ time. We create a list L of segments, which at the beginning is empty, and will contain the segments in S when we are done. For each line ℓ that contains at least $k + 1$ points, we find the extremal points p and q of P contained in ℓ in time $\mathcal{O}(n)$. Then we add the line segment with endpoints p and q to L . Using the algorithm presented by Guibas et al. [11], we can compute all lines containing more than k points in time $\mathcal{O}(\frac{n^2}{k} \log(\frac{n}{k}))$. If there are more than k of such lines, we already know that there is no covering of P with at most k non-crossing segments of P . Indeed such a covering can only exist if there exists a

covering of P with at most k lines. Let us now assume that there are at most k such lines. We add all corresponding segments to L in total time $\mathcal{O}(kn + \frac{n^2}{k} \log(\frac{n}{k}))$. Let us show that the segments in L satisfy properties a), b) and c), although they might still be crossing. First, property a) holds by definition. Moreover property b) holds for all coverings of P with at most k segments because a segment in L containing points p and q also contains all points on the line (p, q) . Finally, property c) also holds trivially for all coverings of P with at most k segments.

We are now going to modify L and make copies of it while maintaining the fact that properties a), b) and c) hold. Our aim is that no two segments in L cross. Let us consider one segment s in L which is crossed by another segment s' in L . We denote by p the intersection of s and s' . We retrieve the points u and v such that uv is the shortest P -segment in s whose relative interior contains p . We do likewise with u' and v' in s' . Observe that not both uv and $u'v'$ can be in a covering of P with non-crossing segments. More generally, in a valid covering, at least one of uv and $u'v'$ is not contained in any segment of the covering. We create one copy of L , and recurse on two cases, one where we assume that uv is not contained in a segment of the covering, and one where we assume that $u'v'$ is not contained in a segment of the covering. Let us assume for now that uv is not contained in a segment of the covering. We keep s' in L , and s' might still be removed at a later step. We remove s from L . The segment s' splits s at p into two sides. Let us denote by x and y the endpoints of s , with u being closer to x than v is. If p is not in P , we consider the segments xu and vy . If p is in P , we consider the segments xp and py . Any of the two new segments that contains more than k points in P is added to L . Indeed property a) holds by definition. Moreover property b) holds because s was in L , and we are assuming that the segment uv is not contained in a segment of the covering. If a segment contains at most k points, we do not add it to L . We claim that property c) still holds. This is because if a point $q \in P$ which lies on a line that contains more than k points is not contained in some segment in L , that means that if a segment t contains q as well as at least k other points in P , then t also contains some segment which we are assuming not to be contained in the covering.

If we obtain more than k segments in L , we stop this branch of the recursion, as we already know that there is no valid covering of P with at most k segments, assuming that uv is not contained in a segment of the covering. We now iterate over all crossing segments in L . We obtain $\mathcal{O}(k)$ segments in L , which are by construction non-crossing. As the depth of the recursion tree is in $\mathcal{O}(k^2)$, the number of leaves is in $\mathcal{O}(2^{k^2})$. We would like to say that each recursion implies the existence of one more segment in a covering with non-crossing segments, but this is a priori not the case. Therefore, if the number of lines containing more than k points is in $\Omega(k)$, we might have to do $\Omega(k^2)$ recursions. We can do the computation in total time $\mathcal{O}(2^{k^2} + kn + \frac{n^2}{k} \log(\frac{n}{k}))$, using the information we preprocessed. If we add to it the running time of the preprocessing, the total running time of the algorithm is in $\mathcal{O}(2^{k^2} + n^4 \log n)$. ◀

The proof of Theorem 14 appears in the full version of the paper [10]. The idea is to fix one valid covering K if it exists, and then to guess in time $\mathcal{O}(2^{k^2})$ the set $S \in \mathcal{F}$ of segments which corresponds to this covering K . Then we can argue by property c) that there are at most k^2 points in P not contained in some segments in S . It remains to guess what are the segments in K covering those points. If some of these segments split a segment in S , then we simply do as in the proof of Lemma 17 and update the set S of segments.

► **Theorem 18.** *It is possible to compute in time $\mathcal{O}(2^{36f^2} f^{42f+1} + n^4 \log n)$ a convex partition of a point set P with at most $24f$ convex faces, where f denotes the minimum number of convex faces required. The same holds when considering convex tilings.*

Proof. We first compute a minimum covering of the inner points in time $\mathcal{O}(2^{s^2} s^{7s+1} + n^4 \log n)$ by applying the algorithm of Theorem 14 for $k = 1, 2, \dots, s$, where s denotes the minimum number of segments required in a covering of the inner points. Then, by 2. of Theorem 6, we obtain in $\mathcal{O}(n^2)$ -time a convex partition with at most $24f$ convex faces. The same holds with convex tilings for the same arguments. As by 1. of Theorem 6, we have $s \leq 6f$, the total running time of the algorithm is as stated. ◀

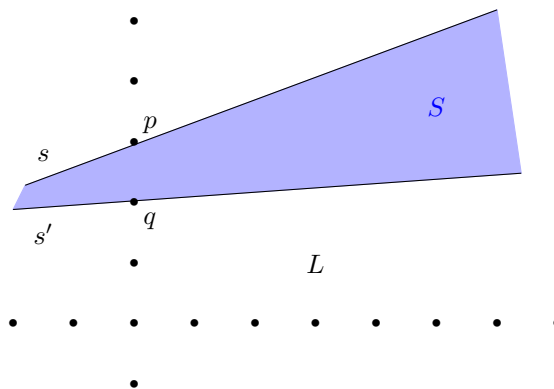
We discuss in the full version of the paper [10] why the membership of CPNCS in FPT does not contradict the W[1]-hardness of Maximum Independent Set in Segment Intersection Graphs shown by Marx [15]. We also discuss why our techniques are not sufficient to obtain an exact FPT algorithm for MCP.

5 NP-hardness of MCP and MCT

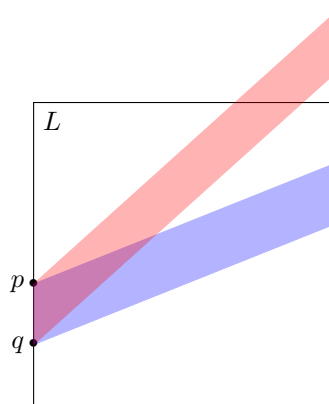
Our proof of NP-hardness of MCP and MCT builds upon gadgets introduced by Lingas [14]. He used them to prove NP-hardness of several decision problems, including *Minimum Convex Partition for Polygons with Holes* and *Minimum Rectangular Partition for Rectilinear Polygons with Holes*. The entire proof appears in the full version of the paper [10]. The idea is to first mimic Lingas' proof. We show how we can embed the rectilinear polygon with holes into a grid Λ of polynomial size. Then we add all edges of the grid outside of the polygon and inside of the holes to the drawing. This gives us a set Φ of unit length segments. We finally replace each unit segment by K collinear points, where K depends polynomially on the size of the input, and obtain a point set P . We show that the convex faces in a minimum convex partition of P have large area, and that if the interior of a convex face F in a convex partition of P intersects a segment in Φ , then the area of F , denoted by $A(F)$, is not large enough. Therefore, the interior of a convex face F in a minimum convex partition of P does not intersect a segment in Φ . From this we can conclude that convex partitions on P behave as if the segments of the polygon were there as constraints. Thus, the reduction works similarly as Lingas'. We present here our key lemma in the proof. It states that if the interior of a convex face F intersects a segment σ in Φ , then F cannot have large area within two cells of Λ on different sides of σ , where large area means larger than $1/K$.

► **Lemma 19.** *Let L and L' be two unit cells in Λ , and let F be a convex polygon whose interior does not contain any point in P . If $A(F \cap L) > 1/K$, and the boundary of F crosses a segment of Φ between L and L' , then $A(F \cap L') < 1/K$.*

Proof. The proof is illustrated in Figure 4. By assumption, F intersects a line segment whose endpoints p and q are at distance $1/K$. Let us consider the two line segments s and s' of the boundary of S that intersect the line ℓ which contains p and q . Assume for contradiction that the lines containing respectively s and s' do not intersect, or intersect on the side of ℓ where L lies. This implies that $F \cap L$ is contained in a parallelogram that has area $1/K$, as illustrated in Figure 5. Indeed such a parallelogram has base $1/K$ and height 1, therefore $A(F \cap L) \leq 1/K$. This shows that the lines containing respectively s and s' intersect on the side of ℓ where L' lies. Using the same arguments as above, this implies $A(F \cap L') < 1/K$. ◀



■ **Figure 4** If $A(S \cap L) > 1/K$, the two lines containing s and s' intersect on the left side.



■ **Figure 5** The area of the parallelograms is $1/K$.

6 Open problems

It would be interesting to have approximation algorithms for MCP, MCT and CPNCS with better ratio than $\mathcal{O}(\log OPT)$. As MCP is not APX-hard unless $NP \subseteq DTIME(2^{polylog n})$ [1], we expect that some improvement can be achieved.

A natural question is to ask whether MCP is FPT with respect to the number of faces in an optimal convex partition, as we have only shown a constant-approximation FPT algorithm. This question is open when having several points on a line is allowed, since otherwise the minimum number of convex faces is linear in the number of inner points.

We have shown that the decision versions of MCP and CPNCS are NP-complete, and that the one of MCT is NP-hard, but the question whether the decision version of MCT is in NP remains open. We also do not know the complexity of MCP and MCT when it is assumed that no three points are collinear.

References

- 1 Sayan Bandyapadhyay, Santanu Bhowmick, and Kasturi Varadarajan. Approximation schemes for partitioning: Convex decomposition and surface approximation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1457–1470. SIAM, 2014. doi:10.1137/1.9781611973730.96.



- 2 Allan S. Barboza, Cid C. de Souza, and Pedro J. de Rezende. Minimum convex partition of point sets. In *Proceedings of International Conference on Algorithms and Complexity*, pages 25–37. Springer, 2019. doi:[10.1007/978-3-030-17402-6_3](https://doi.org/10.1007/978-3-030-17402-6_3).
- 3 Prosenjit Bose, Andrej Brodnik, Svante Carlsson, Erik D Demaine, Rudolf Fleischer, Alejandro López-Ortiz, Pat Morin, and J Ian Munro. Online routing in convex subdivisions. *International Journal of Computational Geometry & Applications*, 12(04):283–295, 2002. doi:[10.1142/S021819590200089X](https://doi.org/10.1142/S021819590200089X).
- 4 Hadrien Cambazard and Nicolas Catusse. An integer programming formulation using convex polygons for the convex partition problem. In *37th International Symposium on Computational Geometry (SoCG 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. doi:[10.4230/LIPIcs.SoCG.2021.20](https://doi.org/10.4230/LIPIcs.SoCG.2021.20).
- 5 Erik Demaine, Sándor Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S. B. Mitchell. CG:SHOP 2020. <https://cgshop.ibr.cs.tu-bs.de/competition/cg-shop-2020>. Accessed: 12/02/2020.
- 6 Adrian Dumitrescu, Sarel Har-Peled, and Csaba D. Tóth. Minimum convex partitions and maximum empty polytopes. In *Proceedings of Scandinavian Workshop on Algorithm Theory*, pages 213–224. Springer, 2012. doi:[10.1007/978-3-642-31155-0_19](https://doi.org/10.1007/978-3-642-31155-0_19).
- 7 Thomas Fevens, Henk Meijer, and David Rappaport. Minimum convex partition of a constrained point set. *Discrete Applied Mathematics*, 109(1-2):95–107, 2001. doi:[10.1016/S0166-218X\(00\)00237-7](https://doi.org/10.1016/S0166-218X(00)00237-7).
- 8 Jesús García-López and Carlos M. Nicolás. Planar point sets with large minimum convex decompositions. *Graphs and Combinatorics*, 29(5):1347–1353, 2013. doi:[10.1007/s00373-012-1181-z](https://doi.org/10.1007/s00373-012-1181-z).
- 9 Magdalene Grantson and Christos Levcopoulos. A fixed parameter algorithm for the minimum number convex partition problem. In *Japanese Conference on Discrete and Computational Geometry*, pages 83–94. Springer, 2004. doi:[10.1007/11589440_9](https://doi.org/10.1007/11589440_9).
- 10 Nicolas Grelier. Hardness and approximation of minimum convex partition. *arXiv preprint*, 2019. arXiv:[1911.07697](https://arxiv.org/abs/1911.07697).
- 11 Leonidas J. Guibas, Mark H. Overmars, and Jean-Marc Robert. The exact fitting problem in higher dimensions. *Computational geometry*, 6(4):215–230, 1996. doi:[10.1016/0925-7721\(95\)00020-8](https://doi.org/10.1016/0925-7721(95)00020-8).
- 12 Christian Knauer and Andreas Spillner. Approximation algorithms for the minimum convex partition problem. In *Proceedings of Scandinavian Workshop on Algorithm Theory*, pages 232–241. Springer, 2006. doi:[10.1007/11785293_23](https://doi.org/10.1007/11785293_23).
- 13 Stefan Langerman and Pat Morin. Covering things with things. *Discrete & Computational Geometry*, 33(4):717–729, 2005. doi:[10.1007/s00454-004-1108-4](https://doi.org/10.1007/s00454-004-1108-4).
- 14 Andrzej Lingas. The power of non-rectilinear holes. In *Proceedings of International Colloquium on Automata, Languages, and Programming*, pages 369–383. Springer, 1982. doi:[10.1007/BFb0012784](https://doi.org/10.1007/BFb0012784).
- 15 Dániel Marx. Parameterized complexity of independence and domination on geometric graphs. In *International Workshop on Parameterized and Exact Computation*, pages 154–165. Springer, 2006. doi:[10.1007/11847250_14](https://doi.org/10.1007/11847250_14).
- 16 Joseph S. B. Mitchell. Approximation algorithms for geometric separation problems. *Technical report, Dept. of Applied Math. and Statistics, State U. of New York at Stony Brook*, 1993. Available at <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.7089&rep=rep1&type=pdf>.
- 17 Víctor Neumann-Lara, Eduardo Rivera-Campo, and Jorge Urrutia. A note on convex decompositions of a set of points in the plane. *Graphs and Combinatorics*, 20(2):223–231, 2004. doi:[10.1007/s00373-004-0555-2](https://doi.org/10.1007/s00373-004-0555-2).
- 18 Franco P. Preparata and Michael I. Shamos. *Computational Geometry*. Springer-Verlag, New York, 1985. doi:[10.1007/978-1-4612-1098-6](https://doi.org/10.1007/978-1-4612-1098-6).

- 19 Toshinori Sakai and Jorge Urrutia. Convex decompositions of point sets in the plane. *arXiv preprint*, 2019. [arXiv:1909.06105](https://arxiv.org/abs/1909.06105).
- 20 Allan Sapucaia, Pedro J. de Rezende, and Cid C. de Souza. Solving the minimum convex partition of point sets with integer programming. *Computational Geometry*, page 101794, 2021. [doi:10.1016/j.comgeo.2021.101794](https://doi.org/10.1016/j.comgeo.2021.101794).
- 21 Andreas Spillner. A fixed parameter algorithm for optimal convex partitions. *Journal of Discrete Algorithms*, 6(4):561–569, 2008. [doi:10.1016/j.jda.2008.07.002](https://doi.org/10.1016/j.jda.2008.07.002).
- 22 Csaba D. Tóth. Binary space partitions for line segments with a limited number of directions. *SIAM Journal on Computing*, 32(2):307–325, 2003. [doi:10.1137/S0097539702403785](https://doi.org/10.1137/S0097539702403785).

Parameterised Partially-Predrawn Crossing Number

Thekla Hamm  

Algorithms and Complexity Group, TU Wien, Austria

Petr Hliněný  

Faculty of Informatics, Masaryk University, Brno, Czech Republic

Abstract

Inspired by the increasingly popular research on extending partial graph drawings, we propose a new perspective on the traditional and arguably most important geometric graph parameter, the *crossing number*. Specifically, we define the *partially predrawn crossing number* to be the smallest number of crossings in any drawing of a graph, part of which is prescribed on the input (not counting the prescribed crossings). Our main result – an FPT-algorithm to compute the partially predrawn crossing number – combines advanced ideas from research on the classical crossing number and so called *partial planarity* in a very natural but intricate way. Not only do our techniques generalise the known FPT-algorithm by Grohe for computing the standard crossing number, they also allow us to substantially improve a number of recent parameterised results for various drawing extension problems.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability; Theory of computation → Computational geometry

Keywords and phrases Crossing Number, Drawing Extension, Partial Planarity, Parameterised Complexity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.46

Related Version *Full Version*: <https://arxiv.org/abs/2202.13635>

Funding *Thekla Hamm*: Supported by the Austrian Science Fund (projects P31336, Y1329, and W1255-N23).

Petr Hliněný: Supported by the Czech Science Foundation, project no. 20-04567S.

1 Introduction

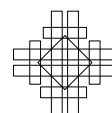
Determining the crossing number, i.e. the smallest possible number of pairwise transverse intersections (called *crossings*) of edges in any drawing, of a graph is among the most important problems in discrete computational geometry. As such its general computational complexity is well-researched: Probably most famously, it is known that graphs with crossing number 0, i.e. planar graphs, can be recognised in polynomial time [27, 20, 28]. Generally, computing the crossing number of a graph is NP-hard, even in very restricted settings [16, 19, 25, 4], and also APX-hard [3]. However there is a fixed-parameter algorithm for the problem, and even one that can compute a drawing of a graph with at most k crossings in time in $\mathcal{O}(f(k)n)$ or decide that its crossing number is larger than k [17, 22].

More recently, so called *graph drawing extension* problems have received increased attention. Instead of being given an entirely abstract graph as an input, here the input is a *partially drawn graph* $\mathcal{P} = (G, \mathcal{H})$, meaning that a subgraph H of the input graph G is given with a fixed drawing \mathcal{H} which must not be changed in the solution. This is motivated by immediate applications in network visualisation [23], as well as a more general line of research in which important computational problems are extended to the setting in which parts of the solution are prescribed which can lead to useful insights for dynamic or divide-and-conquer type algorithms and heuristics [5, 14]. In this context it is natural to define the *partially predrawn crossing number* as the smallest number of pairwise crossings of edges in any



© Thekla Hamm and Petr Hliněný;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 46; pp. 46:1–46:15
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



drawing which coincides with (i.e., extends) the given fixed drawing of the pre drawn skeleton, minus the number of “unavoidable” crossings already contained in the fixed drawing of the skeleton. We name this problem PARTIALLY PRE DRAWN CROSSING NUMBER.

Of course, the problem of computing the partially pre drawn crossing number is more general than the one of computing the classical crossing number (which is captured by the former by simply letting the pre drawn skeleton be empty), and thus the known hardness results for computing the classical crossing number carry over. To the best of our knowledge, the partially pre drawn crossing number problem has so far not been explicitly studied in literature, although, there are papers which study *partially embedded planarity*, i.e. the property of having partially pre drawn crossing number 0, and variants thereof. In particular, similarly to ordinary planarity, partially drawn graphs extendable to planar drawings can be recognised in polynomial time [1], and in analogy to the Kuratowski theorem, there is also a neat list of forbidden “partially drawn minors” (Figure 3) which characterise partially drawn graphs extendable to planar drawings [21].

If one allows a non-zero number of crossings, the only algorithmic results on extending partially drawn graphs with constrained crossings we are aware of are those for scenarios with a few edges or vertices outside of the pre drawn skeleton or/and with a small number of crossings for each edge. We give a brief list of these algorithmic results:

- An algorithm to determine the exact partially pre drawn crossing number of a partially drawn graph in FPT time parameterised by the number of edges which are not fixed by the pre drawn skeleton [6] (the “rigid” case in the paper).
- An algorithm to determine whether there is a 1-planar drawing (or more generally a drawing in which each edge outside of the pre drawn skeleton has at most c crossings) which coincides with the given partial drawing in FPT time parameterised by (c and) the number of edges which are not fixed by the pre drawn skeleton [13, 15].
- An algorithm to determine whether there is a 1-planar drawing which coincides with the given partial drawing in XP time parameterised by the vertex cover size of the edges which are not fixed by the pre drawn skeleton [12].
- An algorithm to determine whether there is a simple drawing in which each edge outside of the pre drawn skeleton has at most c crossings which coincides with the given partial drawing in FPT time parameterised by c and the number the edges which are not fixed by the pre drawn skeleton [15].

We remark that all these parameterised algorithms require the given pre drawn skeleton to be connected, and the last three algorithms are easily adapted to output drawings minimising the number of crossings under the requirement of the respective properties.

Contributions

The foundation of our main contribution is a fixed-parameter algorithm for an exact computation of the partially pre drawn crossing number k of a given partially drawn graph.

► **Theorem 1.1.** *PARTIALLY PRE DRAWN CROSSING NUMBER is in FPT when parameterised by the solution value (i.e., by the number of crossings which are not pre drawn).*

We employ a technique similar to the approach showing fixed-parameter tractability of classical crossing number devised by Grohe [17]. This means we proceed in two phases:

- I. We iteratively reduce the input partially drawn graph \mathcal{P} until we cannot find a large flat grid in it, and so we bound its treewidth by a function of k , or decide that the partially pre drawn crossing number of \mathcal{P} is larger than k . Importantly, each reduction step is guaranteed to preserve the solution value (unless it is $> k$).

- II. We devise an MSO_2 -encoding for the property that any partially drawn graph has the partially predrawn crossing number at most k . The key idea is to encode the predrawn skeleton of the input in a 3-connected planar “frame” which is added to the input partially drawn graph. Using the bounded treewidth of the involved graph with the frame, we then apply Courcelle’s theorem [7] in order to decide this property.

Note that the second step is an interesting result in its own right:

► **Lemma 1.2.** *For every $k \geq 0$ there is an MSO_2 -formula ψ_k such that the following holds. Given a partially drawn graph \mathcal{P} , one can in polynomial time construct a graph G' such that ψ_k is true on G' if and only if the partially predrawn crossing number of \mathcal{P} is at most k . This claim holds also if some edges of \mathcal{P} are marked as “uncrossable” and we compute the crossing number over such drawings extending \mathcal{P} that do not have crossings on the “uncrossable” edges.*

While our high-level approach is similar to Grohe’s [17], in each phase we are faced with some caveats, on which we elaborate in the respective sections, due to the fact that we must respect the given predrawn skeleton and that we have to observe also the treewidth of the derived graph which encodes the predrawn skeleton, i.e. of G' from Lemma 1.2.

In this regard, we also give a concrete example (see Proposition 5.1) of a fundamentally different behaviour of the partially predrawn crossing number compared to the classical one (which can partly explain the difficulties we face in Theorem 1.1, compared to [17]). In a nutshell, we show that for fixed k a partially drawn graph can have arbitrarily many nested cycles which are “critical” for having crossing number $> k$.

Based on the proof of Theorem 1.1 we are also able to give an improved algorithm to determine whether there is a drawing in which each edge outside of the predrawn skeleton has at most c crossings which coincides with the given partial drawing. Specifically we can show the following theorem, where the *partially predrawn c -planar crossing number* of a partially drawn graph \mathcal{P} is as the partially predrawn crossing number above while restricted to only drawings of \mathcal{P} in which each edge outside of the predrawn skeleton has at most c crossings.

► **Theorem 1.3.** *PARTIALLY PREDRAWN c -PLANAR CROSSING NUMBER is in FPT when parameterised by the solution value (i.e., by the number of crossings which are not predrawn).*

Compared to the algorithm given in [13], Theorem 1.3 presents an additional improvement in two important aspects. Not only can our algorithm solve the c -planar drawing extension problem parameterised by the number of new crossings (a less restrictive parameter than the combination of c and $|E(G) \setminus E(H)|$), but we can also handle disconnected initial drawings.

We also can combine our techniques with structural insights from [15] to drop the connectivity requirement on the input in the setting that we want to determine the partially predrawn c -planar crossing number restricted to simple drawings:

► **Theorem 1.4.** *Given a partially drawn graph, one can in FPT time parameterised by c and the number of edges not contained in the predrawn skeleton, decide the minimum number of crossings in a simple drawing which coincides with the given simple partial drawing and in which each edge outside of the predrawn skeleton has at most c crossings.*

Full proofs of the *-marked statements are left for arXiv:2202.13635.



■ **Figure 1** Two drawings of the same graph (solid lines) with the same rotation scheme. However both drawings are not equivalent. All dashed curves need to be mapped without crossing each other or any solid line by any homeomorphism from the left to the right the drawing. This is not possible.

2 Preliminaries

We use standard terminology for undirected simple graphs [9] and assume basic understanding of *parameterised complexity* [8, 10], and of *Courcelle’s theorem together with MSO logic* [2, 7] and treewidth. We refer also to the full preprint paper for additional background on these notions. Regarding embeddings and drawings of graphs we mostly follow [24].

For $r \in \mathbb{N}$, we write $[r]$ as shorthand for the set $\{1, \dots, r\}$.

2.1 Partial graph drawings

A *drawing* \mathcal{G} of a graph G in the Euclidean plane \mathbb{R}^2 is a function that maps each vertex $v \in V(G)$ to a distinct point $\mathcal{G}(v) \in \mathbb{R}^2$ and each edge $e = uv \in E(G)$ to a simple open curve $\mathcal{G}(e) \subset \mathbb{R}^2$ with the ends $\mathcal{G}(u)$ and $\mathcal{G}(v)$. We require that $\mathcal{G}(e)$ is disjoint from $\mathcal{G}(w)$ for all $w \in V(G) \setminus \{u, v\}$. In a slight abuse of notation we often identify a vertex v with its image $\mathcal{G}(v)$ and an edge e with $\mathcal{G}(e)$. Throughout the paper we will moreover assume that: there are finitely many points which are in an intersection of two edges, no more than two edges intersect in any single point other than a vertex, and whenever two edges intersect in a point, they do so transversally (i.e., not tangentially).

The intersection (a point) of two edges is called a *crossing* of these edges. A drawing \mathcal{G} is *planar* (or a *plane graph*) if \mathcal{G} has no crossings, and a graph is *planar* if it has a planar drawing. The number of crossings in a drawing \mathcal{G} is denoted by $\text{cr}(\mathcal{G})$. A drawing \mathcal{G} is *c-planar* (or a *c-plane graph*) if every edge in \mathcal{G} contains at most c crossings, and a graph is *c-planar* if it has a c -planar drawing. The *planarisation* \mathcal{G}^\times of a drawing \mathcal{G} of G is the plane graph obtained from \mathcal{G} by making each crossing point a new degree-4 vertex of \mathcal{G}^\times . The inclusion-maximal connected subsets of the set-complement $\mathbb{R}^2 \setminus \mathcal{G}$ are called the *faces* of \mathcal{G} . For any drawing exactly one of these faces is infinite and referred to as the *outer face*.

A *partial drawing* of a graph G is a drawing of an arbitrary subgraph H of G . A *partially drawn graph* $\mathcal{P} = (G, \mathcal{H})$, with an implicit reference to H , is a graph G together with a partial drawing \mathcal{H} of $H \subseteq G$, and then \mathcal{H} is called the *predrawn skeleton* of (G, \mathcal{H}) . We say that two drawings \mathcal{G}_1 and \mathcal{G}_2 of the same graph G are *equivalent* if there is a homeomorphism of \mathbb{R}^2 onto itself taking \mathcal{G}_1^\times onto \mathcal{G}_2^\times [24]. For connected \mathcal{G}_1^\times and \mathcal{G}_2^\times , this is the same as requiring equal rotation systems and the same outer face. However, for disconnected drawings, [21] in addition to equal rotation systems and outer face it is necessary to specify which faces of each connected component of \mathcal{G}_1^\times contain which other connected components and in which orientation, and match this specification with \mathcal{G}_2^\times (see also Figure 1).

In this setup, we also say that two *partially drawn graphs* are *isomorphic* if there exists an isomorphism which gives an equivalence of their predrawn skeletons.

2.2 Problem definitions

The PARTIALLY PREDRAWN CROSSING NUMBER problem takes as an input a partially drawn graph (G, \mathcal{H}) and an integer q . The task is to decide whether there is a drawing \mathcal{G} of G , the restriction of which to the predrawn skeleton H is equivalent to \mathcal{H} (we can shortly say that \mathcal{G} *extends* \mathcal{H}), such that \mathcal{G} has at most $q + \text{cr}(\mathcal{H})$ crossings. The smallest value of the parameter q for which (G, \mathcal{H}) is a **yes**-instance of PARTIALLY PREDRAWN CROSSING NUMBER is called the *partially predrawn crossing number* of (G, \mathcal{H}) , denoted by $\text{pd-cr}(G, \mathcal{H})$. Note that $\text{pd-cr}(G, \emptyset)$ is the (called classical for distinction) *crossing number* $\text{cr}(G)$ of G .

Likewise, the PARTIALLY PREDRAWN c -PLANAR CROSSING NUMBER problem takes as an input a partially drawn graph (G, \mathcal{H}) and an integer q . The task is to decide whether there is a drawing \mathcal{G} of G in which every edge in $E(G) \setminus E(H)$ has at most c crossings and the restriction of which to H is equivalent to \mathcal{H} , such that \mathcal{G} has altogether at most $q + \text{cr}(\mathcal{H})$ crossings. The smallest q (which may not be defined in general; a trivial example for which q is not defined is given by $c = 1$ and G not 1-planar) for which (G, \mathcal{H}) is a **yes**-instance of PARTIALLY PREDRAWN c -PLANAR CROSSING NUMBER is called the *partially predrawn c -planar crossing number* of (G, \mathcal{H}) .

2.3 A parameterised algorithm for classical crossing number

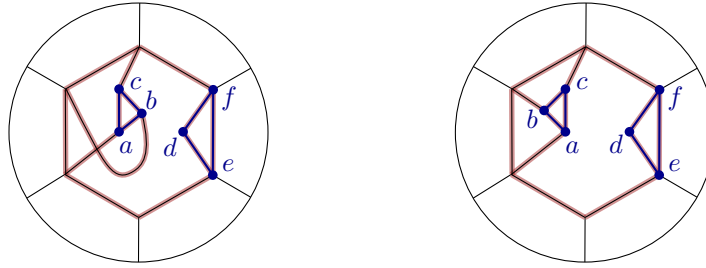
We outline the high-level idea of Grohe’s algorithm [17] to decide the classical crossing number of a graph in FPT time and note some obstacles that we need to overcome. Due to lack of space in the main paper, we leave the complete formal recapitulation together with some supplementary definitions for the full preprint paper.

The algorithm proceeds in two phases.

Phase I – Bounding Treewidth

Consider a graph G in which some edges are marked as “uncrossable”, and the question of whether there is a drawing of G with at most k crossings in which no “uncrossable” edge is crossed for a fixed parameter k . To improve readability, we shortly say that a drawing is *conforming* if no edge marked “uncrossable” is crossed in it. Grohe [17] showed that in polynomial time one can (i) confirm that the answer to this question is no, (ii) find a tree decomposition of G with width bounded in k , or (iii) find a connected planar subgraph $I \subseteq G$ where $|V(I)| \geq 6$ together with a cycle C that is disjoint from $V(I)$ and contains $N(I)$ such that the following holds. If G' arises from G by contracting I to a vertex v_I and additionally marking all edges incident to v_I and all edges of C as “uncrossable”, then any crossing-minimum conforming drawing of G arises from a crossing-minimum conforming drawing of G' by replacing v_I with a planar drawing of $G[V(I) \cup V(C)]$ where the drawing of C is distorted to match that in the drawing of G' and I is drawn in an ε -neighbourhood of v_I . Conversely, every crossing-minimum conforming drawing of G' arises from a crossing-minimum conforming drawing of G by contracting I and placing the resulting vertex on the drawing of some vertex in I .

In the partially drawn setting we can however not simply contract a subgraph I without loosing information about its parts that are potentially fixed by the partial drawing of the instance. In particular, reinserting some unrestricted planar drawing of I can violate the partial drawing (see Figure 2).



■ **Figure 2** Example where predrawn parts (blue) make it impossible to simply insert a planar drawing of I (brown underlay). If the partial drawing is as on the left, I can be drawn planarly as depicted on the right but not while preserving equivalence of the partial drawing (cf. Figure 1).

Phase II – MSO Encoding

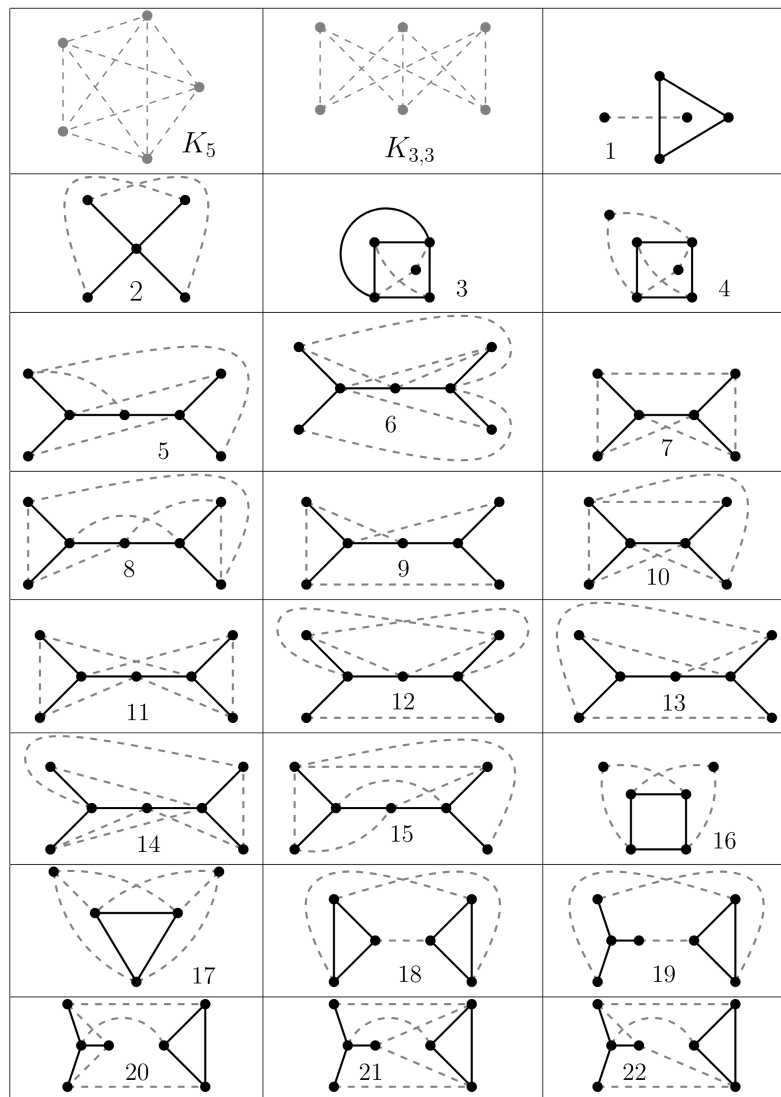
After having reduced G to a graph of treewidth bounded in the desired crossing number, one can apply Courcelle’s theorem to decide whether $\text{cr}(G) \leq k$ for any fixed k . For that it is sufficient to encode in MSO_2 logic the existence of at most k pairs of edges such that, after planarising a hypothetical crossing between the two edges of each pair, the resulting graph is planar. To express planarity, one simply excludes the existence of subdivisions of the two Kuratowski obstructions K_5 and $K_{3,3}$. The task of interpreting the planarisation of hypothetical crossings, “guessed” by existential quantifiers, is a more subtle one. In order to avoid heavy tools of finite model theory here, we can apply the following trick: instead of G , use the graph $G^{(k)}$ which subdivides k -times every edge of G , and “guess” k pairs of the subdivision vertices which are pairwise identified to make the planarisation.

This of course does not carry over easily to the partially drawn setting as the Kuratowski obstructions do not capture the predrawn skeleton shape, i.e., there could be partially drawn graphs with high crossing number and not containing any K_5 or $K_{3,3}$ subdivisions. Here, instead, we will use the corresponding planarity obstructions for partially drawn graphs from [21], described next in Section 2.4. This brings two new complications to be resolved; namely that the list of obstructions is not finite, and that we have to encode the input drawing of the given partially drawn graph in an abstract way which can be “read” by an MSO_2 -formula.

2.4 Characterising partially predrawn planarity

We use the mentioned result of Jelínek, Kratochvíl and Rutter [21] characterising partially predrawn planarity, that is, the question of whether a given partially drawn graph (G, \mathcal{H}) admits a planar drawing which extends \mathcal{H} , by means of forbidding so-called PEG-minors. In this context we assume $\text{cr}(\mathcal{H}) = 0$. The forbidden obstructions are formed by one “easy” infinite family described separately (the *alternating chains*) and a list of 24 specific partially drawn graphs shown in Figure 3. However, since PEG-minors are not suitable for our application, we relax the characterisation of [21] to make a larger finite obstruction set and a simpler-to-handle containment relation (essentially a “partially drawn topological minor”).

A *subdivision* of an edge in a partially drawn graph (G, \mathcal{H}) is the same subdivision in the graph G , which is correspondingly applied to \mathcal{H} if the subdivided edge is from \mathcal{H} . A partially drawn graph (G_1, \mathcal{H}_1) is a (*partially drawn*) *subgraph* of (G, \mathcal{H}) if $G_1 \subseteq G$, $\mathcal{H}_1 \subseteq \mathcal{H}$ and the drawing \mathcal{H}_1 is equivalent to the restriction of \mathcal{H} to \mathcal{H}_1 . Note that in general one may have an edge of G_1 which is predrawn in \mathcal{H} but not in \mathcal{H}_1 .

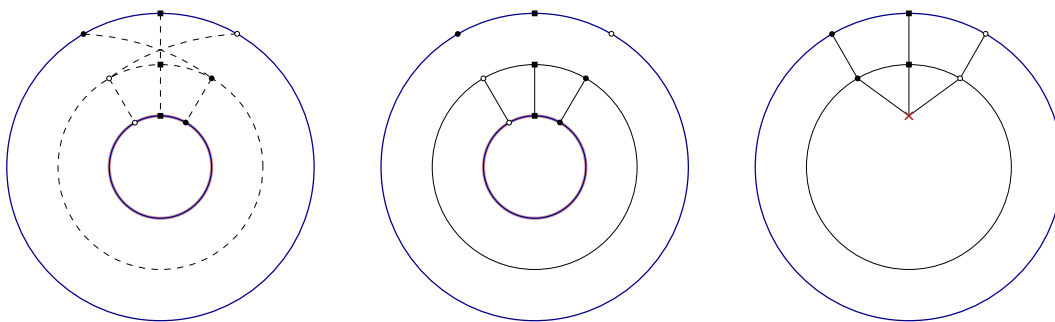


■ **Figure 3** (A picture copied from arXiv:1204.2915v1 with permission of the authors.) The list of 24 partially drawn graphs [21] that are the obstructions (as PEG-minors) for partially drawn graphs which can be extended to planar drawings. The solid black edges and vertices form the predrawn skeleton of the graphs, and dashed edges are the non-fixed ones.

► **Theorem 2.1** (adapted from [21]). *There is a finite family \mathcal{K} of partially drawn graphs such that the following is true. A partially drawn graph $\mathcal{P} = (G, \mathcal{H})$ admits a planar drawing which extends \mathcal{H} if and only if $\text{cr}(\mathcal{H}) = 0$ and the following hold:*

- i. *there is no alternating chain in \mathcal{P} (see the preprint version for the full definition), and*
- ii. *no subdivision of a partially drawn graph from \mathcal{K} is isomorphic to a partially drawn subgraph of \mathcal{P} .*

Briefly put, the family \mathcal{K} from Theorem 2.1 is composed of all graphs obtained from the obstructions (G, \mathcal{H}) in Figure 3 [21] by possible iterative splittings (of vertices of degree > 3 in G) and possible releasing of certain edges from \mathcal{H} . The *splitting* of a vertex v is performed by partitioning the neighbourhood of v into two disjoint sets N_1 and N_2 , and replacing v



■ **Figure 4** Instance (left) cannot be drawn with 0 crossings (same node styles indicate adjacencies to vertices on dashed cycle), but subinstance (middle) consisting only of H (blue), and induced graph on I (brown underlay) and C , as well as subinstance (right) in which I is contracted can.

with two new adjacent vertices v_1 and v_2 such that the neighbourhood of v_1 is $N_1 \cup \{v_2\}$ and the neighbourhood of v_2 is $N_2 \cup \{v_1\}$. The *release* of an edge $f \in E(H)$ from \mathcal{H} is allowed if f is a bridge, i.e. f is not contained in any cycle of H , and is performed as follows: If one end (resp., both ends) of f is of degree > 2 in H , subdivide f once (twice), and denote by f' the edge resulting from f such that both ends of f' are of degree ≤ 2 in H . Then remove f' only from \mathcal{H} (but keep it in G). We leave the details for the full preprint paper.

3 Algorithm for partially predrawn crossing number

Note that, regarding the input partially drawn graph (G, \mathcal{H}) , we may as well assume that \mathcal{H} is a plane graph; otherwise, we replace \mathcal{H} with its planarisation \mathcal{H}^\times (and accordingly adjust G , which formally means to move to the partially drawn graph $((G - E(H)) \cup \mathcal{H}^\times, \mathcal{H}^\times)$). This is sound since neither do we care about the number of crossings prescribed by \mathcal{H} , nor do we have any restrictions on single edges in H , and hence do not care to identify them. Thus, we will assume planar \mathcal{H} throughout the rest of the section, unless we explicitly say otherwise.

3.1 Phase I – Treewidth

To show that we can arrive at an input graph with small treewidth, we prove a statement analogous to Grohe’s iterative contraction for the partially predrawn setting. Approaching this, however, it becomes quite clear that contracting a subgraph I must be treated much more delicately. The role of the cycle C in that case is that it could be treated as an interface to glue together two drawings – any planar drawing of the contracted part and any drawing of G after contraction with at most k crossings in which no “uncrossable” edge is crossed. For actually gluing the parts together, the drawing of C might need to be “flipped” in either of these two drawings. This can create a problem in terms of being equivalent to \mathcal{H} on H . Even if we ensure that each of the two drawings we would potentially like to glue together to a drawing of G are compatible with \mathcal{H} or the contraction of \mathcal{H} , this compatibility is not invariant under flipping C (see e.g. Figure 4).

For this purpose we consider the notion of $(\mathcal{H}, \mathcal{I})$ -*flippability* for C and I . Essentially, we say that C is $(\mathcal{H}, \mathcal{I})$ -flippable in a graph D , if the orientation of C with respect to I in a planar drawing of D that is equivalent to \mathcal{H} on H is not determined by \mathcal{H} . Otherwise C is $(\mathcal{H}, \mathcal{I})$ -unflippable in D . A formal definition that makes use of the non-equivalence of drawing two disconnected triangles described in Figure 1 is given in the full preprint paper. Using this formal definition it can be decided in polynomial time whether a cycle is $(\mathcal{H}, \mathcal{I})$ -flippable in a graph, or not.

To facilitate readability, we say that for a partially drawn graph (G, \mathcal{H}) where some edges of G are marked as “uncrossable”, the drawings of G that we want to consider, are k -crossing conforming. More formally, a k -crossing conforming drawing is a drawing of G with at most $k + \text{cr}(\mathcal{H})$ crossings that is equivalent to \mathcal{H} on the predrawn skeleton H and in which no “uncrossable” edge is crossed. The following key theorem is fully stated and proved in the preprint paper.

- **Theorem 3.1.** *For all $k \in \mathbb{N}$ there exists $w \in \mathbb{N}$, such that given a partially drawn graph (G, \mathcal{H}) in which some edges are marked “uncrossable”, in FPT-time parameterised by k we can*
1. *decide that there is no k -crossing conforming drawing of (G, \mathcal{H}) ; or*
 2. *find a tree decomposition of G of width at most w ; or*
 3. *find an equivalent instance (G', \mathcal{H}') with the property that $|V(G')| < |V(G)|$.*

Sketch of proof. We start by applying the result by Grohe [17] for k with G as input. If the algorithm of [17] decides that the number of crossings in any drawing of G in which no “uncrossable” edge is crossed is more than k times, we can safely return that the same is true for any such drawing that is equivalent to \mathcal{H} on the predrawn skeleton. Similarly, if the algorithm returns a tree decomposition of width at most w , we can return that tree decomposition.

In the last case, the algorithm finds a subgraph $I \subseteq G$ and a cycle C in G as described in Subsection 2.3 for bounding treewidth. We distinguish whether there is a 0-crossing conforming drawing of $(G[V(I) \cup V(C)] \cup H, \mathcal{H})$, or not. Recall that, as we assume \mathcal{H} to be planarised, edges marked as “uncrossable” are irrelevant in this context because no edge should be crossed. Hence deciding whether there is a 0-crossing conforming drawing of $(G[V(I) \cup V(C)] \cup H, \mathcal{H})$ is equivalent to deciding whether $\text{pd-cr}(G[V(I) \cup V(C)] \cup H, \mathcal{H}) = 0$. This can be decided in linear time using the result by Angelini et al. [1].

▷ **Case 1.** There is no 0-crossing conforming drawing of $(G[V(I) \cup V(C)] \cup H, \mathcal{H})$.

In this case we claim that there is no k -crossing conforming drawing of (G, \mathcal{H}) . Assume for a contradiction that there is such a drawing \mathcal{G} . In particular this drawing has at most k crossings and no “uncrossable” edge is crossed in it. Hence, because of the choice of I and C , no edge of $G[V(I) \cup V(C)]$ is crossed in \mathcal{G} . But as there are exactly $\text{cr}(\mathcal{H})$ crossings involving only edges of H in \mathcal{G} , this means that the restriction of \mathcal{G} to $G[V(I) \cup V(C)]$ is a 0-crossing conforming drawing of $(G[V(I) \cup V(C)] \cup H, \mathcal{H})$; a contradiction.

▷ **Case 2.** There is a 0-crossing conforming drawing of $(G[V(I) \cup V(C)] \cup H, \mathcal{H})$.

This is the case in which we attempt to construct an equivalent instance with fewer vertices. Informally speaking, if we find an $(\mathcal{H}, \mathcal{I})$ -flippable cycle C , we will essentially be able to flip any planar drawing of the contracted subgraph to appropriately match the interface in a drawing of G after the contraction. Hence we can simply contract I in G and \mathcal{H} .

If we find a cycle that is $(\mathcal{H}, \mathcal{I})$ -unflippable and the cycle remains unflippable after the contraction of the subgraph is performed, any planar drawing of the contracted subgraph automatically matches the interface in a drawing of G after contraction. Hence we can simply contract I in G and \mathcal{H} .

The last case is that the cycle we find is $(\mathcal{H}, \mathcal{I})$ -unflippable but it seems to be flippable after the contraction of the subgraph is performed. In this case the orientation of the cycle is fixed in any planar drawing of the subgraph I for contraction, but both orientations of the cycle are possible after the contraction is performed. We must therefore appropriately force the orientation of C in the drawing after performing the contraction to match the one which is in fact forced before the contraction. We will do this by extending \mathcal{H} carefully. ◀

We can iteratively apply Theorem 3.1 $\mathcal{O}(|V(G)|)$ times to reduce our instance to a graph of small treewidth. Hence from now on we focus on the case that we are given a partially drawn graph (G, \mathcal{H}) and a tree decomposition of G whose width w is bounded in the inquired crossing number.

This is already a crucial step towards the targeted application of Courcelle’s theorem. However we still need to incorporate the information on the partial drawing \mathcal{H} into a graph structure of small treewidth. For this we will define a *framing* of (G, \mathcal{H}) . Note that even though we assume in this definition \mathcal{H} to be planar, the definition also applies to the general case in which we first planarise \mathcal{H} into \mathcal{H}^\times and correspondingly adjust G .

► **Definition 3.2.** A framing of a partially drawn graph (G, \mathcal{H}) , where \mathcal{H} is a plane graph, is an ordinary (abstract) graph F constructed as follows. See Figure 5. We start with the initial drawing $\mathcal{D} := \mathcal{H}$ and continue by the following steps in order:

1. While the graph of \mathcal{D} is not connected, we iteratively add edges from G to \mathcal{D} that can be inserted in a planar way and which connect two previously disconnected components. If this is no longer possible while the graph is still disconnected, let B be a face of \mathcal{D} incident to more than one connected component. We pick a vertex v on B and connect v to an arbitrary vertex from each component incident to B which does not contain v . We will call all edges added in this step the connector edges (of the resulting framing).
2. We replace each edge $f = uw$ of the drawing \mathcal{D} from Step 1 (including the connector edges) by three internally disjoint paths of length 3 between u and w . We will call these three paths together the framing triplet of f , and denote by \mathcal{D}' the resulting drawing.
3. Around each vertex $v \in V(\mathcal{H}^\times)$ in the drawing \mathcal{D}' from Step 2, we add a cycle on the neighbours of v in \mathcal{D}' in the cyclic order given by \mathcal{D}' . We will call these cycles the framing cycles, and all edges of the resulting planar drawing \mathcal{D}'' the frame edges.
4. Finally, we set $F := \mathcal{D}'' \cup G$ where \mathcal{D}'' is the underlying graph of \mathcal{D}'' from Step 3.

We remark that Step 1 of the construction of a framing F of (G, \mathcal{H}) is not deterministic, and hence a partially drawn graph can admit multiple framings. Note also that possible connector edges introduced in Step 1 are no longer present in resulting F (only their vertices and derived frame triplets are present). Moreover, the most important aspect of Definition 3.2 is that the frame (\mathcal{D}'') defined after Step 3 is a 3-connected planar graph which hence combinatorially captures the drawing \mathcal{H} within the framing F .

As the last step in preparation for applying Courcelle’s theorem we need to show that the framing construction does not considerably increase the treewidth:

► **Lemma 3.3.*** Let F be a framing of a partially drawn graph (G, \mathcal{H}) , and $G^\circ = (G - E(\mathcal{H})) \cup \mathcal{H}^\times$. Then $\text{tw}(F) \in \mathcal{O}(16^{k+1} \text{tw}(G^\circ) / \log(\text{tw}(G^\circ)))$, where $k = \text{pd-cr}(G, \mathcal{H})$.

3.2 Phase II – MSO₂-encoding

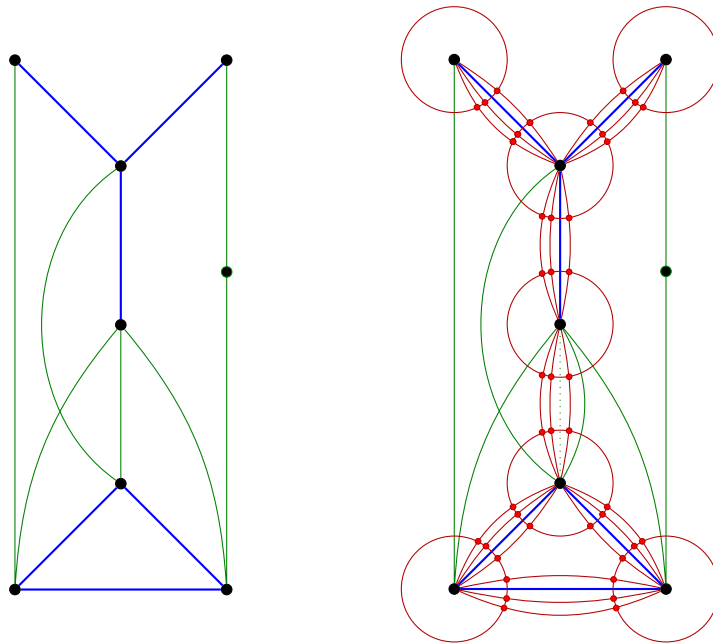
Our aim now is to prove key Lemma 1.2. In closer detail, we are first going to show:

► **Lemma 3.4.*** Let $\mathcal{P}_1 = (G_1, \mathcal{H}_1)$ be a partially drawn graph where \mathcal{H}_1 is plane. There exists an MSO₂-formula σ , depending on \mathcal{P}_1 , such that the following is true:

- For any partially drawn graph $\mathcal{P}_2 = (G_2, \mathcal{H}_2)$ with plane \mathcal{H}_2 and any framing \bar{G}_2 of \mathcal{P}_2 we have that $\bar{G}_2 \models \sigma$, if and only if some subdivision of \mathcal{P}_1 is a partially drawn subgraph of \mathcal{P}_2 .

To combinatorially characterise the partially drawn subgraph containment, we use Definition 3.2 and the following concept of a “framing-aware” minor. Considering framings \bar{G}_1 of (G_1, \mathcal{H}_1) and \bar{G}_2 of (G_2, \mathcal{H}_2) , we say that \bar{G}_1 is a *framing topological minor* of \bar{G}_2 if there is a topological-minor embedding of \bar{G}_1 into \bar{G}_2 which additionally satisfies

- every edge of G_1 (resp., of \mathcal{H}_1) is mapped into a path of G_2 (resp., of \mathcal{H}_2),



■ **Figure 5** (Definition 3.2) A *framing* of a partially drawn graph (G, \mathcal{H}) : the graph is on the left, such that the predrawn skeleton \mathcal{H} is drawn with thick blue edges and the remaining edges of $E(G) \setminus E(H)$ are in green. The framing of (G, \mathcal{H}) on the right has the frame edges drawn in red; for every edge of \mathcal{H} and for the chosen one connector edge between the two components of H , we get a framing triplet, and for every vertex of \mathcal{H} a framing cycle.

- every framing cycle in \tilde{G}_1 is mapped into a corresponding framing cycle in \tilde{G}_2 ,
- whenever an edge $f \in E(H_1)$ is mapped into a path $P_f \subseteq H_2$, the framing triplet of f in \tilde{G}_1 is embedded (as three internally-disjoint paths) in the union of the framing cycles and triplets of the internal vertices and edges of P_f in \tilde{G}_2 , and
- the analogous condition (as the previous point) applies also to framing triplets of the connector edges of \tilde{G}_1 , which are embedded in \tilde{G}_2 .

See Figure 6 for a natural illustration of this concept.

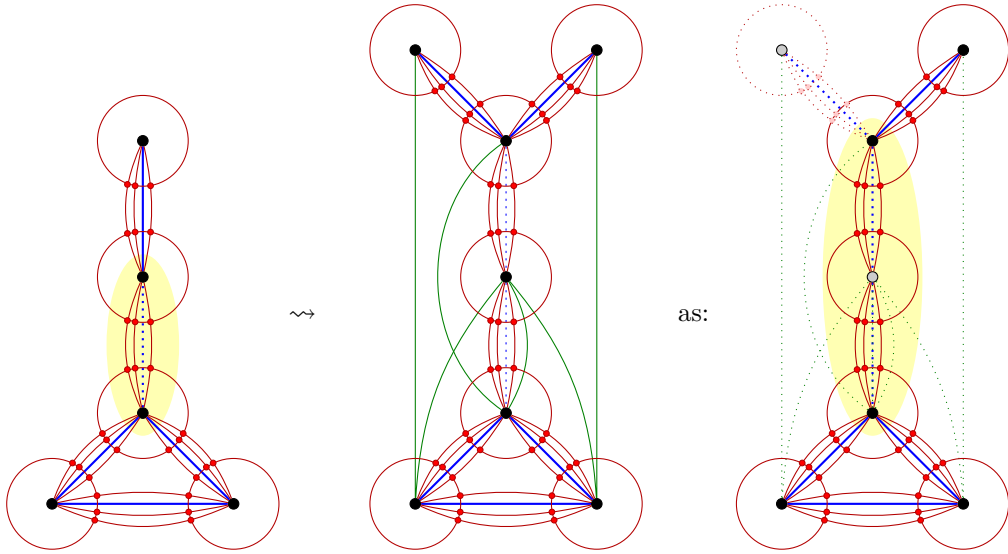
However, to state the desired characterisation we still need to technically generalise Definition 3.2 to an *extended framing* of a partially drawn graph (G, \mathcal{H}) which, informally, allows us to use possible additional connector vertices and arbitrary connector edges between the components of \mathcal{H} . See the preprint paper for all details.

► **Lemma 3.5.*** *Let $\mathcal{P}_1 = (G_1, \mathcal{H}_1)$ and $\mathcal{P}_2 = (G_2, \mathcal{H}_2)$ be partially drawn graphs where \mathcal{H}_1 and \mathcal{H}_2 are plane. Let \tilde{G}_2 be a framing of \mathcal{P}_2 . Then some subdivision of \mathcal{P}_1 is a partially drawn subgraph of \mathcal{P}_2 , if and only if there exists an extended framing \tilde{G}_1 of \mathcal{P}_1 such that \tilde{G}_1 is a restricted topological minor of \tilde{G}_2 .*

We now finish a proof sketch of Lemma 3.4 easily. Let \mathcal{F} be the finite set of all distinct extended framings of \mathcal{P}_1 . Using Lemma 3.5, we may write the formula $\sigma \equiv \bigvee_{\tilde{G}_1 \in \mathcal{F}} \sigma[\tilde{G}_1]$ where $\tilde{G}_2 \models \sigma[\tilde{G}_1]$ routinely expresses that \tilde{G}_1 is a framing topological minor of \tilde{G}_2 (this description uses auxiliary precomputed labels distinguishing the types of edges in \tilde{G}_2).

We also need to address the other kind of obstruction in Theorem 2.1 with the following:

- **Lemma 3.6.*** *There exists an MSO₂-formula τ such that the following is true:*
- *For any partially drawn graph $\mathcal{P}_2 = (G_2, \mathcal{H}_2)$ and any framing \tilde{G}_2 of \mathcal{P}_2 we have that $\tilde{G}_2 \models \tau$, if and only if there exists an alternating chain in \mathcal{P}_2 .*



■ **Figure 6** An illustration of the framing topological minor relation; the framing \bar{G}_1 (of the 5-vertex partially drawn graph (G_1, \mathcal{H}_1)) on the left is embedded in the framing \bar{G}_2 (of the 7-vertex graph (G_2, \mathcal{H}_2)) in the middle, and this embedding is emphasised as a topological minor in the picture on the right. Notice that the framing triplet in \bar{G}_1 highlighted in the left picture with yellow background is mapped (as three internally disjoint red paths) into a union of two framing triplets plus the intermediate framing cycle in \bar{G}_2 , as highlighted with yellow background in the picture on the right.

Now we can sketch a proof of the key Lemma 1.2 which we reformulate slightly for clarity:

► **Lemma 3.7** (Lemma 1.2). *For every $k \geq 0$ there is an MSO_2 -formula ψ_k such that the following holds. Given a partially drawn graph \mathcal{P} , with some edges of \mathcal{P} marked as “uncrossable”, one can in polynomial time construct a graph G' such that $G' \models \psi_k$ if and only if there exists a k -crossing conforming drawing of \mathcal{P} .*

Sketch of proof. Recall that we may assume \mathcal{H} to be a plane graph. We first give a rough outline of what we want to achieve and then sketch the core steps of the proof.

The graph G' will be based on a framing (as used above). Imagine a conforming drawing \mathcal{G} of G (extending \mathcal{H}) with $\text{cr}(\mathcal{G}) = k$ and its planarisation \mathcal{G}^\times . If we were able to “guess”, within the formula ψ_k , the additional k vertices (those of \mathcal{G}^\times) making the crossings, then we would finish by checking partially predrawn planarity of the result (i.e., of the guessed \mathcal{G}^\times). Using Theorem 2.1, the latter would follow by an application of Lemmas 3.4 and 3.6.

Specifically, for the task of “guessing the crossings”, we subdivide each edge of \mathcal{P} which is not marked as “uncrossable” by k new vertices, called *auxiliary vertices* of this partially drawn subdivision $\mathcal{P}_0 = (G_0, \mathcal{H}_0)$ of \mathcal{P} . A subdivision clearly does not change the crossing number; $\text{cr}(\mathcal{P}) = \text{cr}(\mathcal{P}_0)$. Then we interpret “guessing a crossing” in \mathcal{P}_0 as picking (with existential quantifiers in ψ_k) a pair $r'_1, r''_1 \in V(G_0) \setminus V(G)$ of auxiliary vertices such that not both r'_1 and r''_1 are from edges of H , and identifying $r'_1 = r''_1$. Let $\mathcal{P}_0[r'_1 = r''_1]$ denote the graph after such an identification. Note that since we do not identify auxiliary pairs from two edges of H , the following holds – if \bar{G}_0 is a framing of \mathcal{P}_0 , then $\bar{G}_0[r'_1 = r''_1]$ is a graph isomorphic to the corresponding framing of $\mathcal{P}_0[r'_1 = r''_1]$.

We let $G' = \bar{G}_0$ be a framing of $\mathcal{P}_0 = (G_0, \mathcal{H}_0)$. Let $\mathbf{r}' = (r'_i : i \in [k])$ and $\mathbf{r}'' = (r''_i : i \in [k])$ be two k -tuples of vertex variables (which are used to specify the k identifications of vertex pairs in $\mathcal{P}_0[\mathbf{r}' = \mathbf{r}'']$). We write the desired formula as

$$\psi_k \equiv \exists \mathbf{r}', \mathbf{r}'' \left(\bigwedge_{r, s \in \mathbf{r}' \cup \mathbf{r}''} r \neq s \wedge \bigwedge_{i \in [k]} \chi(r'_i, r''_i) \wedge \psi'_k[\mathbf{r}', \mathbf{r}''] \right),$$

where $\chi(r'_i, r''_i)$ checks that r'_i, r''_i are auxiliary vertices and not both coming from edges of H (using precomputed labels of the auxiliary vertices). The formula $\psi'_k[r', r'']$ then tests whether the partially drawn graph $\mathcal{P}_0[r' = r'']$ admits a planar drawing extending \mathcal{H}_0 . This is a technical task based on Lemmas 3.4 and 3.6, and we leave full details for the preprint paper. \blacktriangleleft

Finally, we summarise how **Theorem 1.1** follows from the previous claims. Given a partially drawn graph (G, \mathcal{H}) and an integer $k > 0$, we first make \mathcal{H} planarised. Then, using Theorem 3.1, we either conclude that $\text{pd-cr}(G, \mathcal{H}) > k$, or we iteratively reduce the input to an equivalent instance (G', \mathcal{H}') with the same solution value k . Moreover, using also Lemma 3.3, we have that the tree-width of any framing \tilde{G}' of (G', \mathcal{H}') is bounded in terms of k . We can hence efficiently decide whether $\text{pd-cr}(G', \mathcal{H}') \leq k$ using Courcelle's theorem applied with the formula ψ_k from Lemma 3.7 to a framing \tilde{G}' of (G', \mathcal{H}') .

(*) We can also observe that the FPTruntime of this procedure is $\mathcal{O}(f(k) \cdot |V(G)|^3)$.

4 Restricting crossings per edge

Next we outline some nice consequences of our techniques for previously considered drawing extension settings. Firstly, we are able to trivially modify our FPT-algorithm for PARTIALLY PREDRAWN CROSSING NUMBER by additionally encoding the fact that in a solution every edge in $E(G) \setminus E(H)$ has at most c crossings by introducing k auxiliary vertices for each edge in $E(H)$, but only $\min\{c, k\}$ auxiliary vertices for each edge in $E(G) \setminus E(H)$ in the proof of Lemma 3.7. This immediately gives us Theorem 1.3 restated from above.

► **Theorem 1.3.** *PARTIALLY PREDRAWN c -PLANAR CROSSING NUMBER is in FPT when parameterised by the solution value (i.e., by the number of crossings which are not predrawn).*

Another closely related problem that has been considered in literature asks for the smallest number of non-predrawn crossings in a *simple* drawing that coincides with the given partially drawn graph, in which each edge in $E(G) \setminus E(H)$ has at most c crossings. I.e., compared to PARTIALLY PREDRAWN c -PLANAR CROSSING NUMBER we only allow drawings in which no pair of edges crosses more than once (crossings between adjacent edges can always be avoided). The difficulty for our approach here is that we need to record the information of which edges in \mathcal{H}^\times correspond to the same edge in the non-planarised predrawn skeleton H (this part can be handled by an MSO₂-formula with help of special edge labels, cf. [15]), and more importantly to keep this information, even during our iterative reduction of G and \mathcal{H}^\times described in Section 3.1. The latter seems to be a deep problem, not easy to overcome and a good direction for continuing research.

Nevertheless, using the more restrictive parameterisation by $|E(G) \setminus E(H)| + c$ (which also naturally bounds the crossing number), we are able to give an improvement on the best known result in [15]: finding the least number of crossings in a simple drawing which coincides with the given partial drawing and in which each edge outside of the predrawn skeleton has at most c crossings in FPT-time. The known result assumes that the planarised predrawn skeleton is connected, an assumption that we can easily drop using our MSO₂-encoding in combination with a crucial structural lemma which we adapt from [15] to “stitch” together relevant edges in \mathcal{H}^\times that correspond to the same edge in H . This improvement over [15] results in **Theorem 1.4** stated in the Introduction.

5 Conclusion

To summarise, we have shown that some algorithmic results for the classical crossing-number can be extended to the partially pre drawn setting, similarly to the respective planarity question [1]. However, what can we say about structural properties of the partially pre drawn crossing number?

For instance, what can we say about the minimal graphs of a certain crossing-number value? We call a partially drawn graph $\mathcal{P} = (G, \mathcal{H})$ *k-crossing-critical* if the partially pre drawn crossing number of \mathcal{P} is at least k , but this crossing number drops down below k after deleting any edge, pre drawn or not, from \mathcal{P} (alternatively, one may also include removing any edge from H while keeping it in G to the definition). We have recently gotten a complete rough asymptotical characterisation of classical k -crossing-critical graphs [11], but here we see an important difference in behaviour. For classical k -crossing-critical graphs, optimal drawings (i.e. those achieving the minimum number of crossings) can never contain a collection of edge-disjoint cycles drawn nested in each other and of size arbitrarily large compared to k (this is implicit in [18] or [11]). In contrast to that, we provide:

► **Proposition 5.1.*** *For each $k \geq 8$ and $m > 0$, there exists a partially drawn graph $\mathcal{P} = (G, \mathcal{H})$ such that \mathcal{P} is k -crossing-critical and that an optimal (with minimum crossings) drawing of \mathcal{P} extending \mathcal{H} contains at least m vertex-disjoint nested cycles from $G - E(H)$.*

Consequently, even a rough characterisation of partially drawn k -crossing-critical graphs is a widely open question worth further investigation. Unfortunately, already at the starting point of this track we lack a good analogue of the result [26], saying that a k -crossing-critical graph has its crossing number bounded in terms of k , whose proof simply breaks down in the partially pre drawn setting. Having a result like [26] in the pre drawn setting we could, as a first step, adapt the arguments from Section 3 to prove that partially drawn k -crossing-critical graphs have treewidth bounded in terms of k .

References

- 1 Patrizio Angelini, Giuseppe Di Battista, Fabrizio Frati, Vít Jelínek, Jan Kratochvíl, Maurizio Patrignani, and Ignaz Rutter. Testing planarity of partially embedded graphs. *ACM Trans. Algorithms*, 11(4), April 2015. doi:10.1145/2629341.
- 2 Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991. doi:10.1016/0196-6774(91)90006-K.
- 3 Sergio Cabello. Hardness of approximation for crossing number. *Discrete Comput. Geom.*, 49(2):348–358, March 2013.
- 4 Sergio Cabello and Bojan Mohar. Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM J. Comput.*, 42(5):1803–1829, January 2013.
- 5 Katrin Casel, Henning Fernau, Mehdi Khosravian Ghadikolaei, Jérôme Monnot, and Florian Sikora. On the complexity of solution extension of optimization problems. *Theoretical Computer Science*, 2021. doi:10.1016/j.tcs.2021.10.017.
- 6 Markus Chimani and Petr Hliněný. Inserting multiple edges into a planar graph. In *SoCG*, volume 51 of *LIPICs*, pages 30:1–30:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.
- 7 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.
- 8 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

- 9 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 10 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.
- 11 Zdenek Dvořák, Petr Hliněný, and Bojan Mohar. Structure and generation of crossing-critical graphs. In *SoCG*, volume 99 of *LIPICs*, pages 33:1–33:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- 12 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending nearly complete 1-planar drawings in polynomial time. In Javier Esparza and Daniel Král', editors, *MFCS 2020*, volume 170 of *LIPICs*, pages 31:1–31:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.MFCS.2020.31.
- 13 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending partial 1-planar drawings. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *ICALP 2020*, volume 168 of *LIPICs*, pages 43:1–43:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ICALP.2020.43.
- 14 Eduard Eiben, Robert Ganian, Thekla Hamm, and O-joung Kwon. Measuring what matters: A hybrid approach to dynamic programming with treewidth. *Journal of Computer and System Sciences*, 121:57–75, 2021. doi:10.1016/j.jcss.2021.04.005.
- 15 Robert Ganian, Thekla Hamm, Fabian Klute, Irene Parada, and Birgit Vogtenhuber. Crossing-optimal extension of simple drawings. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *ICALP 2021*, volume 198 of *LIPICs*, pages 72:1–72:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ICALP.2021.72.
- 16 Michael R. Garey and David S. Johnson. Crossing number is NP-complete. *SIAM J. Algebr. Discrete Methods*, 4(3):312–316, September 1983.
- 17 Martin Grohe. Computing crossing numbers in quadratic time. *J. Comput. Syst. Sci.*, 68(2):285–302, 2004. doi:10.1016/j.jcss.2003.07.008.
- 18 César Hernández-Vélez, Gelasio Salazar, and Robin Thomas. Nested cycles in large triangulations and crossing-critical graphs. *J. Comb. Theory, Ser. B*, 102(1):86–92, 2012.
- 19 Petr Hliněný. Crossing number is hard for cubic graphs. *Journal of Comb. Theory, Ser. B*, 96(4):455–471, 2006. doi:10.1016/j.jctb.2005.09.009.
- 20 John Hopcroft and Robert Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, October 1974. doi:10.1145/321850.321852.
- 21 Vít Jelínek, Jan Kratochvíl, and Ignaz Rutter. A Kuratowski-type theorem for planarity of partially embedded graphs. *Computational Geometry*, 46(4):466–492, 2013. SoCG 2011. doi:10.1016/j.comgeo.2012.07.005.
- 22 Ken-ichi Kawarabayashi and Bruce Reed. Computing crossing number in linear time. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 382–390. Association for Computing Machinery, 2007. doi:10.1145/1250790.1250848.
- 23 Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, 6(2):183–210, 1995. doi:10.1006/jv1c.1995.1010.
- 24 Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins series in the mathematical sciences. Johns Hopkins University Press, 2001.
- 25 Michael J. Pelsmajer, Marcus Schaefer, and Daniel Stefankovic. Crossing numbers of graphs with rotation systems. *Algorithmica*, 60(3):679–702, 2011.
- 26 Robert B. Richter and Carsten Thomassen. Minimal graphs with crossing number at least k . *J. Comb. Theory, Ser. B*, 58(2):217–224, 1993.
- 27 Klaus Wagner. Über eine Eigenschaft der ebenen Komplexe. *Mathematische Annalen*, 114:570–590, 1937.
- 28 Shih Wei-Kuan and Hsu Wen-Lian. A new planarity test. *Theoretical Computer Science*, 223(1):179–191, 1999. doi:10.1016/S0304-3975(98)00120-0.

Approximation Algorithms for Maximum Matchings in Geometric Intersection Graphs

Sariel Har-Peled ✉ 

Department of Computer Science, University of Illinois,
201 N. Goodwin Avenue, Urbana, IL 61801, USA

Everett Yang ✉

Department of Computer Science, University of Illinois,
201 N. Goodwin Avenue, Urbana, IL 61801, USA

Abstract

We present a $(1-\varepsilon)$ -approximation algorithms for maximum cardinality matchings in disk intersection graphs – all with near linear running time. We also present an estimation algorithm that returns $(1 \pm \varepsilon)$ -approximation to the size of such matchings – this algorithm runs in linear time for unit disks, and $O(n \log n)$ for general disks (as long as the density is relatively small).

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Matchings, disk intersection graphs, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.47

Related Version *Full Version:* <https://arxiv.org/abs/2201.01849>

Funding *Sariel Har-Peled:* Work on this paper was partially supported by a NSF AF award CCF-1907400.

Acknowledgements The authors thank the anonymous referees for their detailed comments.

1 Introduction

Geometric intersection graphs

Given a set of n objects U , its intersection graph, \mathcal{I}_U , is the graph where the vertices correspond to objects in U and there is an edge between two vertices if their corresponding objects intersect. Such graphs can be dense (i.e., have $\Theta(n^2)$ edges), but they have a linear size representation. It is natural to ask if one can solve problems on such graphs more efficiently than explicitly represented graphs.

Maximum matchings

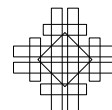
Computing maximum cardinality matchings is one of the classical problems on graphs (surprisingly, the algorithm to solve the bipartite case goes back to work by Jacobi in the mid 19th century). The fastest combinatorial algorithm (ignoring polylog factors) seems to be the work by Gabow and Tarjan [7], running in $O(m\sqrt{n})$ time where m is the number of edges in the graph. Harvey [11] and Mucha and Sankowski [15] provided algorithms based on algebraic approach that runs in $O(n^\omega)$ time, where $O(n^\omega)$ is the fastest time known for multiplying two $n \times n$ matrices. Currently, the fastest known algorithm for matrix multiplication has $\omega \approx 2.3728596$, but it is far from being practical.



© Sariel Har-Peled and Everett Yang;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 47; pp. 47:1–47:13
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Matchings for planar graphs and disk intersection graphs

Mucha and Sankowski [16] adapted their algebraic technique for planar graphs (specifically using separators), getting running time $O(n^{\omega/2}) \approx O(n^{1.17})$. Yuster and Zwick [17] adapted this algorithm for graphs with excluded minors.

Maximum matchings in geometric intersection graphs

Bonnet et al. [3] studied the problem for geometric intersection graphs. For simplicity of exposition, we describe their results in the context of disk intersection graphs. Given a set n disks with maximum density ρ (i.e., roughly the maximum number of disks covering a point in the plane), they presented an algorithm for computing maximum matchings with running time $O(\rho^{3\omega/2} n^{\omega/2}) \approx O(\rho^{3.5} n^{1.17})$. This compares favorably with the naive algorithm of just plugging such graphs into the algorithm of Gabow and Tarjan, which yields running time $O(m\sqrt{n}) = O(\rho n^{3/2})$. If the ratio between the smallest disk and largest disk is at most Φ , they presented an algorithm with running time $O(\Phi^{12\omega} n^{\omega/2})$. Note that the running time of all these algorithms is super linear in n .

Approximate maximum matchings

It is well known that it is enough to augment along paths of length up to $O(1/\varepsilon)$ if one wants $(1 - \varepsilon)$ -approximate matchings. For bipartite graphs this implies that one need to run $O(1/\varepsilon)$ rounds of paths finding stage of the bipartite matching algorithm of Hopcroft and Karp [12]. Since such a round takes $O(m)$ time, this readily leads to an $(1 - \varepsilon)$ -approximate bipartite matching algorithm in this case. The non-bipartite case is significantly more complicated, and the weighted case is even more difficult. Nevertheless, Duan and Pettie [5] presented an algorithm with running time $O(m\varepsilon^{-1} \log \varepsilon^{-1})$ which provides $(1 - \varepsilon)$ -approximation to the maximum weight matching in non-bipartite graph.

Density and approximate matchings

For a set of objects U in \mathbb{R}^d , the density of an object is the number of bigger objects in the set intersecting it. The *density* of the set of objects is the maximum density of the objects. The density is denoted by ρ , and the premise is that for real world inputs it would be small. The intersection graph of such objects when ρ is a constant are known as *low density graphs*, have some nice properties, such as having separators. See [9] and references therein. In particular, for a set of fat objects, the density and the maximum depth (i.e., the maximum number of object covering any point) are roughly the same. It is well known that low density graphs are sparse and have $O(\rho n)$ edges, where n is the number of objects in the set. Since one can compute the intersection graph in $O(n \log n + \rho n)$ time, and plug it into the algorithm of Duan and Pettie [5], it follows that an approximation algorithm with running time $O(n \log n + (\rho n/\varepsilon) \log(1/\varepsilon))$. See Section 4.2 for details. Thus, the challenge is to get better running times than this baseline.

1.1 Our results

Our purpose here is to develop near linear time algorithms for approximate matchings for the unit disk graph and the general disk graph cases. Our results are summarized in Table 1. Note, in this paper, we assume the input to our algorithms to be a set of disks.

1. Unit disk graph.
 - a. Greedy matching. We show in Section 3.1 a linear time algorithm for the case of unit disks graph – this readily provides a $1/2$ -approximation to the maximum matching. The algorithm uses a simple grid to “capture” intersections, and then use the locality of the grid to find intersection with the remaining set of disks.
 - b. $(1 - \varepsilon)$ -approximation. In Section 3.2 we show how to get a $(1 - \varepsilon)$ -approximation. The running time can be bounded by $O((n/\varepsilon^2) \log(1/\varepsilon))$. If the diameter of union of disks is at most Δ , then the running time is $O(n + (\Delta^2/\varepsilon^2) \log(1/\varepsilon))$.
 - c. $(1 - \varepsilon)$ -estimation. Surprisingly, one can do even better – we show in Section 3.3 how to use importance sampling to get $(1 \pm \varepsilon)$ -approximation (in expectation) to the size of the maximum matching in time $O(n + \text{poly}(\log n, 1/\varepsilon))$.
2. Disk graph. The general disk graph case is more challenging.
 - a. Greedy matching. The greedy matching algorithm can be implemented in $O(n \log n)$ time using sweeping, see Lemma 15 (this algorithm works for any nicely behaved shapes).
 - b. Approximate bipartite case. Here, we are given two sets of disks, and consider only intersections across the sets as edges. This case can be solved using range searching data-structures as was done by Efrat et al. [6] – they showed how to implement a round of the bipartite matching algorithm of Hopcroft and Karp [12] using $O(n)$ queries. It is folklore that running $O(1/\varepsilon)$ rounds of this algorithm leads to a $(1 - \varepsilon)$ -approximation algorithm. Coupling this with the data-structure of Kaplan et al. [13] readily leads to a near linear approximation algorithm in this case, see Section 4.3.
 - c. $(1 - \varepsilon)$ -approximation algorithm. Surprisingly, approximate general matchings can be reduced to bipartite matchings via random coloring. Specifically, one can compute $(1 - \varepsilon)$ -approximate matchings using $2^{O(1/\varepsilon)} \log n$ invocations of approximate bipartite matchings algorithm mentioned above. This rather neat idea is due to Lotker et al. [14] who used in the context of parallel matching algorithms. This leads to a near linear time algorithm for $(1 - \varepsilon)$ -approximate matchings for disk intersection graphs, see Section 4.4 for details. The running time of the resulting algorithm is $2^{O(1/\varepsilon)} n \log^{O(1)} n$. We emphasize that this algorithm assumes nothing about the density of the input disks.
 - d. $(1 - \varepsilon)$ -estimation. One can get an $O(n \log n)$ time estimation algorithm in this case, but one needs to assume that the input disks have “small” density ρ . Specifically, one computes a separator hierarchy and then use importance sampling on the patches, as to estimate the sampling size. The details require some care, see Section 4.6. The resulting algorithm has running time $O(n \log n)$ if the density is $o(n^{1/9})$.
3. General shapes. Somewhat surprisingly, almost all our results extends in a verbatim fashion to intersection graphs of general shapes. We need some standard assumptions about the shapes:
 - a. the boundary of any pair of shapes intersects only a constant number of times,
 - b. these intersections can be computed in constant time,
 - c. one can compute the x -extreme and y -extreme points in a shape in constant time,
 - d. one can decide in constant time if a point is inside a shape, and
 - e. the boundary of a shape intersects any line a constant number of times, and they can be computed in constant time.

For fat shapes of similar size, we also assume the diameters of all the shapes are the same up to a constant factor, and any object o contains a disk of radius $\Omega(\text{diam}(o))$.

■ **Table 1** Results. The (★) indicates the result works only for disks.

Shape	Quality	Running time	Ref	Comment
Unit disks Fat shapes of similar size	1/2	$O(n)$	Lemma 9	Greedy
	$1 - \varepsilon$	$O\left(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon}\right)$	Lemma 11	
		$O\left(n + (\Delta^2/\varepsilon^2) \log \frac{1}{\varepsilon}\right)$	Remark 12	$\Delta = \text{Diam. disks}$
$1 \pm \varepsilon$	$O(n + \varepsilon^{-6} \log^2 n)$	Theorem 14	Estimation	
Unit disks	$1 - \varepsilon$	$O\left(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon}\right)$	Lemma 11	Approximate
Disks	$1 - \varepsilon$	$O\left((n/\varepsilon) \log^{11} n\right)$	Lemma 18 (★)	Bipartite
	$1 - \varepsilon$	$O(2^{O(1/\varepsilon)} n \log^{12} n)$	Theorem 20 (★)	General
Shapes	1/2	$O(n \log n)$	Lemma 15	Greedy
	$1 - \varepsilon$	$O(n \log n + \frac{m}{\varepsilon} \log \frac{1}{\varepsilon})$	Lemma 16	$m : \# \text{ edges}$
	$1 - \varepsilon$	$O(n \log n + \frac{n\rho}{\varepsilon} \log \frac{1}{\varepsilon})$	Lemma 17	$\rho : \text{Density}$
	Exact	$O(n \log n)$	Lemma 23	Matching size $O(n^{1/8})$
	$1 \pm \varepsilon$	$O(n \log n + \rho^9 \varepsilon^{-19} \log^2 n)$	Theorem 28	Estimation

Since the modification needed to make the algorithms work for the more general cases are straightforward, we describe the algorithms for disks.

The full version of the paper is available on the [arXiv](#) [10] and includes all missing details/proofs.

2 Preliminaries

2.1 Notations

For a graph G , let \mathcal{M}_G^* denote the maximum cardinality matching in G . Its size is denoted by $m^* = m^*(G) = |\mathcal{M}_G^*|$. For a graph $G = (V, E)$, and a set $X \subseteq V$ the *induced subgraph* of G over X is $G|_X = (X, \{uv \in E \mid u, v \in X\})$. For a set Z , let $G - Z$ denote the graph resulting from G after deleting from it all the vertices of Z . Formally, $G - Z$ is the graph $G|_{V \setminus Z}$.

► **Definition 1.** For a set of objects \mathcal{U} , the *intersection graph* of \mathcal{U} , denoted by $\mathcal{I}_{\mathcal{U}}$, is the graph having \mathcal{U} as its set of vertices, and there is an edge between two objects $o, g \in \mathcal{U}$ if they intersect. Formally,

$$\mathcal{I}_{\mathcal{U}} = (\mathcal{U}, \{og \mid o, g \in \mathcal{U} \text{ and } o \cap g \neq \emptyset\}).$$

For a point $p \in \mathbb{R}^2$, and a set of disks \mathcal{D} , let

$$\mathcal{D} \sqcap p = \{\circ \in \mathcal{D} \mid p \in \circ\}$$

be the set of disks of \mathcal{D} that contain p . Note, that the intersection graph $\mathcal{I}_p = \mathcal{I}_{\mathcal{D} \sqcap p}$ is a clique.

► **Definition 2.** Consider a set of disks \mathcal{D} . A set $\mathcal{D}' \subseteq \mathcal{D}$ is an *independent set* (or simply independent) if no pair of disks of \mathcal{D}' intersects.

2.2 Low density and separators

The following is standard by now, see Har-Peled and Quanrud [9] and references therein.

► **Definition 3.** A set of objects U in \mathbb{R}^d (not necessarily convex or connected) has **density** ρ if any object o (not necessarily in U) intersects at most ρ objects in U with diameter equal or larger than the diameter of o . The minimum such quantity is denoted by $\text{density}(U)$. A graph that can be realized as the intersection graph of a set of objects U in \mathbb{R}^d with density ρ is ρ -**dense**. The set U is **low density** if $\rho = O(1)$.

► **Definition 4.** Let $G = (V, E)$ be an undirected graph. Two sets $X, Y \subseteq V$ are **separate** in G if

1. X and Y are disjoint, and
2. there is no edge between the vertices of X and the vertices of Y in G .

For a constant $\zeta \in (0, 1)$, a set $Z \subseteq V$ is a ζ -**separator** for a set $U \subseteq V$, if $U \setminus Z$ can be partitioned into two separate sets X and Y , with $|X| \leq \zeta|U|$ and $|Y| \leq \zeta|U|$.

► **Lemma 5** ([9]). Let U be a set of n objects in \mathbb{R}^d with density ρ . One can compute, in expected linear time, a sphere \mathbb{S} that intersects in expectation $\tau = O(\rho + \rho^{1/d}n^{1-1/d})$ objects of U . The sphere is computed by picking uniformly its radius from some range of the form $[\alpha, 2\alpha]$. Furthermore, the total number of objects of U strictly inside/outside \mathbb{S} is at most ζn , where ζ is a constant that depends only on d . Namely, the intersection graph \mathcal{I}_U has a separator of size τ formed by all the objects of U intersecting \mathbb{S} .

2.3 Importance sampling

Importance sampling is a standard technique for estimating a sum of terms. Assume that for each term in the summation, one can quickly get a coarse estimate of its value. Furthermore, assume that better estimates are possible but expensive. Importance sampling shows how to sample terms in the summation, then acquire a better estimate *only for the sampled terms*, to get a good estimate for the full summation. In particular, the number of samples is bounded independently of the original number of terms, depending instead on the coarseness of the initial estimates, the probability of success, and the quality of the final output estimate.

► **Lemma 6** ([2]). Let $(\mathcal{H}_1, w_1, e_1), \dots, (\mathcal{H}_r, w_r, e_r)$ be given, where \mathcal{H}_i 's are some structures, and w_i and e_i are numbers, for $i = 1, \dots, r$. Every structure \mathcal{H}_i has an associated weight $\bar{w}(\mathcal{H}_i) \geq 0$ (the exact value of $\bar{w}(\mathcal{H}_i)$ is not given to us). In addition, let $\xi > 0$, γ , b , and M be parameters, such that:

1. $\forall i \quad w_i, e_i \geq 1$,
2. $\forall i \quad e_i/b \leq \bar{w}(\mathcal{H}_i) \leq e_i b$, and
3. $\Gamma = \sum_i w_i \cdot \bar{w}(\mathcal{H}_i) \leq M$.

Then, one can compute a new sequence of triples $(\mathcal{H}'_1, w'_1, e'_1), \dots, (\mathcal{H}'_t, w'_t, e'_t)$, that also complies with the above conditions, such that the estimate $Y = \sum_{i=1}^t w'_i \bar{w}(\mathcal{H}'_i)$ is a multiplicative $(1 \pm \xi)$ -approximation to Γ , with probability $\geq 1 - \gamma$. The running time of the algorithm is $O(r)$, and size of the output sequence is $t = O(b^4 \xi^{-2} (\log \log M + \log \gamma^{-1}) \log M)$.

► **Remark 7.**

- (A) The algorithm of Lemma 6 does not use the entities \mathcal{H}_i directly at all. In particular, the \mathcal{H}'_i s are just (reweighed) copies of some original structures. The only thing that the above lemma uses is the estimates e_1, \dots, e_r and the weights w_1, \dots, w_r .
- (B) We are going to use Lemma 6, with $\xi = O(\varepsilon)$, $\gamma = 1/n^{O(1)}$, $b = 2$, and $M = n$. As such, the size of the output list is $L_{\text{len}} = O(\varepsilon^{-2} \log^2 n)$

2.4 Background on matchings

For a graph G , a **matching** is a set $\mathcal{C} \subseteq E(G)$ of edges, such that no pair of them not share an endpoint. A matching that has the largest cardinality possible for a graph G , is a **maximum matching**. Given a graph G and a matching \mathcal{C} on G , an **alternating path** is a path with edges that alternate between matched edges (i.e., edges that are in \mathcal{C}) and unmatched edges (i.e., edges in $E(G) \setminus \mathcal{C}$). If both endpoints of an alternating path are unmatched (i.e., **free**), then it is an **augmenting path**. In the following, let \mathcal{M}^* denote a maximum cardinality matching in G , and let $m^* = m^*(G) = |\mathcal{M}^*|$ denote its size. For $\beta \in [0, 1]$, a matching $\mathcal{B} \subseteq E(G)$ is an **β -matching** (or β -approximate matching) if $|\mathcal{B}| \geq \beta m^*$. The set of vertices covered by the matching \mathcal{B} is denoted by $V(\mathcal{B}) = \bigcup_{uv \in \mathcal{B}} \{u, v\}$.

The *length* of a path is the number of its edges.

The following claim is well known, see [10].

► **Lemma 8.** *For any $\varepsilon \in (0, 1)$, if $|\mathcal{C}| < (1 - \varepsilon)|\mathcal{M}^*|$, then there are at least $(\varepsilon/2)|\mathcal{M}^*|$ disjoint augmenting paths of \mathcal{C} , each of length at most $4/\varepsilon$.*

3 Approximate matchings for unit disk graph

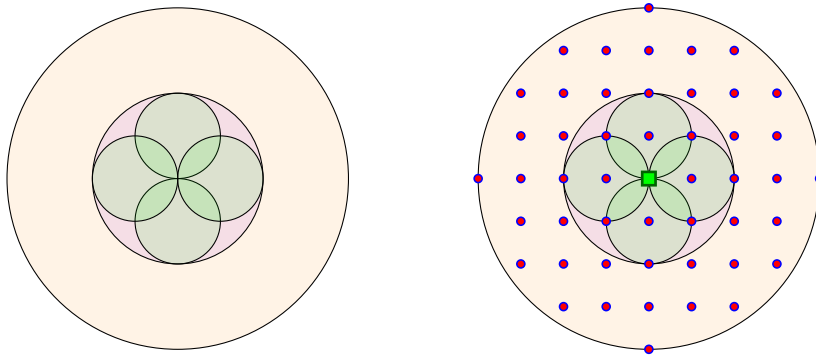
3.1 Greedy maximal matching

In a graph G , the greedy maximal matching can be computed by repeatedly picking an edge of G , adding it to the matching, and removing the two vertices of the edges from G . We do this repeatedly until no edges remain. The resulting **greedy matching** is a maximal matching, and every maximal matching is a $1/2$ -approximation to the maximum matching. To avoid the maximum/maximal confusion, we refer to such a matching as a greedy matching.

► **Lemma 9.** *Let \mathcal{D} be a set of n unit disks in the plane, where a unit disk has radius one. One can compute, in $O(n)$ time, a $(1/2)$ -approximate matching for $\mathcal{I}_{\mathcal{D}}$, where $\mathcal{I}_{\mathcal{D}}$ is the intersection graph of the disks of \mathcal{D} .*

Proof. For every disk, compute all the integral grid points that it covers. Every disk covers at least one, and at most five grid points. We use hashing to compute for every grid point the disks that covers it. These lists can be computed in $O(n)$ time overall. Next, for every grid point that stores more than one disk, scan it, and break it into pairs, where every pair is reported as a matching edge, and the two disks involved are removed.

By the end of this process, we computed a partial matching \mathcal{C} , and we have a set \mathcal{D}' of leftover disks that are not matched yet. The disks of \mathcal{D}' cover every integral grid point at most once. Using the hash table one can look for intersections – for every grid point that is active (i.e., has one disk of \mathcal{D}' covering it), the algorithm lookup in the hash table any disk that covers any of the 8 neighboring grid points. Each such neighboring point offers one disk that might intersect the current disk. If we find an intersecting pair, the algorithm outputs it (removing the two disks involved). This requires $O(1)$ time per active grid point, and linear time overall. At the end of this process, all the remaining disks are disjoint, implying that the computed matching is maximal and thus a $(1/2)$ -approximation to the maximum matching. ◀



■ **Figure 1** A tower can interact with at most 48 other towers.

3.2 $(1 - \varepsilon)$ -approximation

► **Lemma 10.** *Let \mathcal{D} be a set of n unit disks, and let $\varepsilon \in (0, 1)$ be a parameter. Then, one can compute, in $O((n/\varepsilon^2) \log(1/\varepsilon))$ time, an $(1 - \varepsilon)$ -matching in $\mathcal{I}_{\mathcal{D}}$, where $\mathcal{I}_{\mathcal{D}}$ is the intersection graph of \mathcal{D} .*

If the diameter of $\cup \mathcal{D}$ is Δ , then the running time is $O(n + (\Delta^2/\varepsilon^3) \log(1/\varepsilon))$.

Proof. Using a unit grid, the algorithm computes for each disk in \mathcal{D} a grid point that it contains, and register the disk with this point. Using hashing this can be done in $O(n)$ time overall. For a grid point p , let $\ell(p)$ be the list of disks that are registered with it. Let p_1, \dots, p_τ be the points with non-empty lists.

For a point p_i , the graph $\mathcal{I}_{\mathcal{D} \cap p_i}$ is the **tower** of p_i . Consider a maximum matching \mathcal{M}^* of $\mathcal{I}_{\mathcal{D}}$. An edge $uv \in \mathcal{M}^*$ is a **cross edge** if u and v belong to two different towers. Observe that if there are two cross edges between two towers in a matching, then we can exchange them by two edges internal to the two towers, preserving the size of the matching – this observation is due to Bonnet et al. [3]. As such, we can assume that there is at most one cross edge between any two towers in the maximum matching \mathcal{M}^* . In addition, any tower can have edges only with towers in its neighborhood – specifically, two towers might have an edge between them, if the distance between their centers is at most 4. As such, the number of cross edges in \mathcal{M}^* is at most $24\tau = 48\tau/2$, as each tower interacts with at most 48 other towers, see Figure 1.

If a tower has more than (say) $200/\varepsilon$ disks in it, then we add the greedy matching in the tower to the output, and remove all the disks in the tower. This yields a matching of size at least $100/\varepsilon$, that destroys at most 48 additional edges from the optimal matching. Thus, it is sufficient to compute the approximate matching in the residual graph. We repeat this process till all the remaining towers have at most $O(1/\varepsilon)$ disks in them. Let \mathcal{D}' be the remaining set of disks – by the bounded depth, each disk intersects at most $O(1/\varepsilon)$ other disks, and the intersection graph of $G = \mathcal{I}_{\mathcal{D}'}$ has $|E(G)| = O(n/\varepsilon)$ edges, and can be computed in $O(n/\varepsilon)$ time. Using the algorithm of Duan and Pettie [5] on $\mathcal{I}_{\mathcal{D}'}$, computing a $(1 - \varepsilon/2)$ -approximate matching takes $O(|E(G)|\varepsilon^{-1} \log \varepsilon^{-1}) = O(n\varepsilon^{-2} \log \varepsilon^{-1})$ time. It is straightforward to verify that this matching, together with the greedy matching of the “tall” towers yields that desired $(1 - \varepsilon)$ -approximation.

As for the running for the case that the diameter Δ is relatively small – observe that after the cleanup step, there are at most $O(\Delta^2)$ towers, and each tower contains $O(1/\varepsilon)$ disks (each disk intersects $O(1/\varepsilon)$ disks). As such, the residual graph has at most $O(\Delta^2/\varepsilon^2)$ edges, and the running time of the Duan and Pettie [5] algorithm is $O((\Delta^2/\varepsilon^3) \log(1/\varepsilon))$. ◀

Using a reduction of Bonnet et al. [3], we can reduce the residual graph even further, resulting in a slightly faster algorithm.

► **Lemma 11.** *Let \mathcal{D} be a set of n unit disks, and let $\varepsilon \in (0, 1)$ be a parameter. One can compute, in $O((n/\varepsilon) \log(1/\varepsilon))$ time, an $(1 - \varepsilon)$ -matching in $\mathcal{I}_{\mathcal{D}}$.*

► **Remark 12.** If the set of disks of \mathcal{D} has diameter Δ , then the cleanup stage reduces the number of disks to $O(\Delta^2/\varepsilon)$. Then one can invoke the algorithm of Lemma 11. The resulting algorithm has running time $O(n + (\Delta^2/\varepsilon^2) \log(1/\varepsilon))$.

► **Remark 13.** The above algorithm can be modified to work in similar time for shapes of similar size – the only non-trivial step is computing the intersection graph H . This involves taking all the shapes in a tower, and its neighboring towers, computing their arrangement, and extracting the intersection pairs. If this involves ν shapes, then this takes $O(\nu^2)$ time. Each shape would be charged $O(\nu^2/\nu)$ amortized time, which results in $O(n/\varepsilon)$ time, as $\nu = O(1/\varepsilon)$. The rest of the algorithm remains the same.

3.3 Matching size estimation

► **Theorem 14.** *Let \mathcal{D} be a set of n unit disks, and let $\varepsilon \in (0, 1)$ be a parameter. One can output a number Z , such that $(1 - \varepsilon)m^* \leq \mathbb{E}[Z]$ and $\mathbb{P}[Z < (1 + \varepsilon)m^*] \geq 1 - 1/n^{O(1)}$, where m^* is the size of the maximum matching in $\mathcal{I}_{\mathcal{D}}$, where $\mathcal{I}_{\mathcal{D}}$ is the intersection graph of \mathcal{D} . The running time of the algorithm is $O(n + \varepsilon^{-6} \log \varepsilon^{-1} \log^2 n)$.*

Proof. We randomly shift a grid of size length $\psi = \lceil 32/\varepsilon \rceil$ over the plane, by choosing a random point $p \in [0, \psi]^2$. Formally, the (i, j) th cell in this grid is $p + (i\psi, j\psi) + [0, \psi]^2$, where i, j are integers. For a pair of unit disks that intersect, with probability $\geq 1 - \varepsilon/2$ they both fall into the interior of a single grid cell. As such, throwing away all the disks that intersect the boundaries of the shifted grid, the remaining set of disks \mathcal{D}' , in expectation, has a matching of size at least $(1 - \varepsilon/8)m^*$.

For a grid cell \square in this shifted grid, let \mathcal{D}_{\square} be the set of disks of \mathcal{D} that are fully contained in \square . A grid cell \square is *active* if \mathcal{D}_{\square} is not empty. Let \mathcal{B} be the set of active grid cells. For a cell $\square \in \mathcal{B}$, let m_{\square}^* be the size of the maximum matching in $\mathcal{I}_{\mathcal{D}_{\square}}$. Using the algorithm of Lemma 9, compute in $O(n)$ time overall, for all $\square \in \mathcal{B}$, a number e_{\square} such that $m_{\square}^*/2 \leq e_{\square} \leq m_{\square}^*$.

The task at hand is to estimate the sum $\sigma = \sum_{\square \in \mathcal{B}} m_{\square}^*$, where $\sigma \leq m^*$ and $\mathbb{E}[\sigma] \geq (1 - \varepsilon/8)m^*$. To this end, we use importance sampling to reduce the number of terms in the summation of σ that need to be evaluated. Each term m_{\square}^* is 1/2-approximated by e_{\square} , and thus applying the algorithm of Lemma 6, to these approximation, with $\xi = \varepsilon/32$, $b = 2$, $M = n$, and $\gamma = 1/n^{10}$, we get that

$$t = O(b^4 \xi^{-2} (\log \log M + \log \gamma^{-1}) \log M) = O(\varepsilon^{-2} \log^2 n)$$

terms need to be evaluated (exactly if possible, but a $(1 - \varepsilon/16)$ -approximation is sufficient) to get $1 \pm \varepsilon/8$ estimate for σ . For each such cell, we apply the algorithm of Remark 12, to get $(1 - \varepsilon/16)$ -approximation. For a cell \square this takes $O(|\mathcal{D}_{\square}| + (1/\varepsilon^4) \log(1/\varepsilon))$ time. Summing over all these t cells, the running time is $O(n + (t/\varepsilon^4) \log(1/\varepsilon))$. ◀

4 Approximate maximum matching for general disks

4.1 The greedy algorithm

The following $1/2$ -approximation algorithm works (with the same running time) for any simply connected shapes that are well-behaved.

► **Lemma 15.** *Let \mathcal{D} be a set of n disks in the plane. One can compute a greedy matching \mathcal{C} for $\mathcal{I}_{\mathcal{D}}$ in $O(n \log n)$ time. This matching \mathcal{C} is a $1/2$ -approximation – that is, $|\mathcal{C}| \geq m^*/2$, where m^* is the size of the maximum cardinality matching in $\mathcal{I}_{\mathcal{D}}$.*

4.2 Approximation algorithm when the graph is sparse

► **Lemma 16.** *Let \mathcal{D} be a set of n disks in the plane such that the intersection graph $\mathcal{I}_{\mathcal{D}}$ has m edges. For a parameter $\varepsilon \in (0, 1)$, one can compute, in $O(n \log n + (m/\varepsilon) \log(1/\varepsilon))$ time, an $(1 - \varepsilon)$ -matching in $\mathcal{I}_{\mathcal{D}}$.*

Proof. Computing the vertical decomposition of the arrangement $\mathcal{A}(\mathcal{D})$ can be done in $O(n \log n + m)$ randomized time, using randomized incremental construction [4], as the complexity of $\mathcal{A}(\mathcal{D})$ is $O(n + m)$. This readily generates all the edges that arise out of pairs of disks with intersecting boundaries.

The remaining edges are created by one disk being enclosed completely inside another disk. One can perform a DFS on the dual graph of this arrangement, such that whenever visiting a trapezoid, the traversal maintains the set of disks that contains it. This takes time linear in the size of the arrangement, since the list of disks containing a point changes by at most one element between two adjacent faces. Now, whenever visiting a vertical trapezoid that on its non-empty vertical wall on the left contains an extreme right endpoint of a disk \circ , the algorithm reports all the disks that contains this face, as having an edge with \circ . Since every edge is generated at most $O(1)$ times by this algorithm, it follows that its overall running time is $O(n \log n + m)$.

Now that we computed the intersection graph, we apply the algorithm of Duan and Pettie [5]. This takes $O((m/\varepsilon) \log(1/\varepsilon))$ time, and computes the desired matchings. ◀

The above is sufficient if the intersection graph is sparse, as is the case if the graph is low density.

► **Lemma 17.** *Let \mathcal{D} be a set of n disks in the plane with density ρ . For a parameter $\varepsilon \in (0, 1)$, one can $(1 - \varepsilon)$ -approximate the maximum matching in $\mathcal{I}_{\mathcal{D}}$ in $O(n \log n + \frac{n\rho}{\varepsilon} \log \frac{1}{\varepsilon})$ time.*

Proof. The smallest disk in \mathcal{D} intersects at most ρ other disks of \mathcal{D} . Removing this disk and repeating this argument, implies that $\mathcal{I}_{\mathcal{D}}$ has at most ρn edges. The result now readily follows from Lemma 16. ◀

4.3 The bipartite case

Consider computing maximum matching when given two sets of disks $\mathcal{D}_1, \mathcal{D}_2$, where one considers only intersections between disks that belong to different sets – that is the bipartite case. Efrat et al. [6] showed how to implement one round of Hopcroft-Karp algorithm using $O(n)$ dynamic range searching operations on a set of disks. Using the (recent) data-structure

47:10 Approximation Algorithms for Max Matchings in Geometric Intersection Graphs

of Kaplan et al. [13], one can implement this algorithm. Each operation on the dynamic disks data-structure takes $O(\log^{11} n)$ time. If our purpose is to get an $(1 - \varepsilon)$ -approximation, we need to run this algorithm $O(1/\varepsilon)$ times, so that all paths of length $O(1/\varepsilon)$ get augmented, resulting in the following.

► **Lemma 18.** *Given sets $\mathcal{D}_1, \mathcal{D}_2$ at most n disks in the plane, one can $(1 - \varepsilon)$ -approximate the maximum matching in the bipartite graph*

$$\mathcal{I}_{\mathcal{D}_1, \mathcal{D}_2} = (\mathcal{D}_1 \cup \mathcal{D}_2, \{\circ_1 \circ_2 \mid \circ_1 \in \mathcal{D}_1, \circ_2 \in \mathcal{D}_2, \text{ and } \circ_1 \cap \circ_2 \neq \emptyset\}).$$

in $O((n/\varepsilon) \log^{11} n)$ time. Any augmenting path for this matching has length at least $4/\varepsilon$.

4.4 Approximate matching via reduction to the bipartite case

We use a reduction, due to Lotker et al. [14], of approximate general matchings to the bipartite case.

4.4.1 The Algorithm

The input is a set \mathcal{D} of n disks, and a parameter $\varepsilon \in (0, 1)$. The algorithm maintains a matching \mathcal{C} in $\mathcal{I}_{\mathcal{D}}$. Initially, this matching can be the greedy matching. Now, the algorithm repeats the following $O(c_\varepsilon \log n)$ times, where $c_\varepsilon = 2^{8/\varepsilon}$:

***i*th iteration:** Randomly color the disks of \mathcal{D} by two colors (say 1 and 2), and let $\mathcal{D}_1, \mathcal{D}_2$ be the resulting partition. Remove from \mathcal{D}_1 any pair of disks \circ_1, \circ_2 such that $\circ_1 \circ_2$ is in the current matching \mathcal{C} . Do the same to \mathcal{D}_2 . Let \mathcal{C}'_i be edges of \mathcal{C} that appear in $H_i = \mathcal{I}_{\mathcal{D}_1, \mathcal{D}_2}$. Using Lemma 18, find an $(1 - \varepsilon/16)$ -approximate maximum matching in H_i , and let \mathcal{C}''_i be this matching. Augment \mathcal{C} with the augmenting paths in $\mathcal{C}'_i \oplus \mathcal{C}''_i$.

The intuition behind this algorithm is that this process would compute all the augmenting paths of \mathcal{C} of length (say) $\leq 4/\varepsilon$, which implies that the resulting matching is the desired approximation.

4.4.2 Analysis

► **Lemma 19.** *The above algorithm outputs a matching of size $\geq (1 - \varepsilon)m^*$, with probability $\geq 1 - 1/n^{O(1)}$.*

4.4.3 The result

► **Theorem 20.** *Let \mathcal{D} be a set of n disks in the plane, and $\varepsilon \in (0, 1)$ be a parameter. One can compute a matching in $\mathcal{I}_{\mathcal{D}}$ of size $\geq (1 - \varepsilon)m^*$, in $O(2^{8/\varepsilon} n \log^{12} n)$ time, where m^* is the cardinality of the maximum matching in $\mathcal{I}_{\mathcal{D}}$. The algorithm succeeds with high probability.*

► **Remark 21.** Note, that the above algorithm does not work for fat shapes (even of similar size), since the range searching data-structure of Kaplan et al. [13] can not to be used for such shapes.

4.5 Algorithm for the case the maximum matching is small

If $n_{\mathcal{C}} = |\mathcal{C}|$ is small (say, polylogarithmic), it turns out that one can compute the *maximum* matching exactly in near linear time.

► **Lemma 22.** *For a set X of n disks, and any constant $\delta \in (0, 1)$, one can preprocess X , in $O(n^{3+\delta} \log n)$ time, such that given a query disk \circ , the algorithm outputs, in $O(\log n)$ time, a pointer to a (unique) list containing all the disks intersecting the query disk.*

► **Lemma 23.** *Let \mathcal{D} be a set of n disks in the plane. Then, in $O(n \log n)$ time, one can decide if $m^*(\mathcal{D}) = O(n^{1/8})$, and if so compute and output this maximum matching.*

4.6 Estimation of matching size using separators

The input is a set \mathcal{D} of n disks in the plane with density ρ (if the value of ρ is not given, it can be approximated in near linear time [1]). Our purpose here is to $(1 - \varepsilon)$ -estimate the size of the maximum matching in \mathcal{D} in near linear time. Since we can check (and compute it) if the maximum matching is smaller than $n^{1/8}$ by Lemma 23, in $O(n \log n)$ time, assume that the matching is bigger than that.

4.6.1 Preliminaries

► **Lemma 24** ([8]). *Let p be a point in the plane, and let r be a random number picked uniformly in an interval $[\alpha, 2\alpha]$. Let \mathcal{H} be a set of interior disjoint disks in the plane. Then, the expected number of disks of \mathcal{H} that intersects the circle $\circ = \circ(p, r)$, that is centered at p and has radius r , is $O(\sqrt{|\mathcal{H}|})$.*

4.6.2 Algorithm idea and divisions

A natural approach to our problem is to break the input set of disks into small sets, and then estimate the maximum matching size in each one of them. The problem is that for this to work, we need to partition the disks participating in the optimal matching, as this matching can be significantly smaller than the number of input disks. Since we do not have the optimal matchings, we would use a proxy to this end – the greedy matching. The algorithm recursively partitions it using a random cycle separator provided by Lemma 5. We then partition the disks into three sets – inside the cycle, intersecting the cycle (i.e., the separator), and outside the cycle. The algorithm continues this partition recursively on the in/out sets, forming a partition hierarchy.

► **Remark 25.** For a set generated by this partition, its **boundary** is the set of all disks that intersect it and are not in the set. The algorithm maintains the property that for such a set with t disks, the number of its boundary vertices is bounded by $O(\rho + \sqrt{\rho t})$. This can be ensured by alternately separating for cardinality of the set, and for the cardinality of the boundary vertices, see [9] and references therein for details. For simplicity of exposition we assume this property holds, without going into the low level details required to ensure this.

4.6.3 The algorithm

The input is a set \mathcal{D} of n disks in the plane with density ρ , and parameters $\varepsilon \in (0, 1)$. The algorithm computes the greedy matching, denoted by \mathcal{C} , using Lemma 15. If this matching is smaller than $O(n^{1/8})$, then the algorithm computes the maximum matching using Lemma 23, and returns it.

Otherwise, the algorithm partitions the disks of $\mathcal{H} = \mathcal{V}(\mathcal{C})$ recursively using separators, creating a separator hierarchy as described above. Conceptually, a subproblem here is a region R in the plane formed by the union of some faces in an arrangement of circles (i.e., the separators used in higher level of the recursion). Assume the algorithm has the sets of disks $\mathcal{H}_{\subseteq R} = \{\circ \in \mathcal{H} \mid \circ \subseteq R\}$ and $\mathcal{D}_{\subseteq R} = \{\circ \in \mathcal{D} \mid \circ \subseteq R\}$ at hand. The algorithm computes a separator of $\mathcal{H}_{\subseteq R}$, computes the relevant sets for the children, and continues recursively on the children. Thus, for a node u in this recursion tree, there is a corresponding region $R(u)$, a set of active disks $\mathcal{H}_u = \mathcal{H}_{\subseteq R(u)}$, and $\mathcal{D}_u = \mathcal{D}_{\subseteq R(u)}$.

The recursion stops the construction in node u if $|\mathcal{H}_u| \leq b$, where

$$b = c_3 \rho / \varepsilon^2,$$

and c_3 is some sufficiently large constant. This implies that this recursion tree has $U = O(m^*/b)$ leaves.

If a disk of \mathcal{D} intersects some separator cycles then it is added to the set of “lost” disks \mathcal{L} . The hierarchy maps every disk of $\mathcal{D} \setminus \mathcal{L}$ to a leaf. As such, for every leaf u of the separator tree, there is an associated set \mathcal{D}_u of disks stored there. All these leaf sets, together with \mathcal{L} , form a disjoint partition of \mathcal{D} .

The algorithm now computes for every leaf set a greedy matching, using Lemma 15. Let e_v be the size of this matching. Let Ξ be the set of all leaf nodes. The algorithm next $(1 \pm \varepsilon/4)$ -estimates $\sum_{v \in \Xi} m^*(\mathcal{D}_v)$, using importance sampling, with the estimates $e_v \leq m^*(\mathcal{D}_v) \leq 2e_v$, for all v . Using, Lemma 6, this requires computing $(1 - \varepsilon/8)$ -approximate maximum matching for

$$t = O(2^4 \varepsilon^{-2} (\log \log n + \log n) \log n) = O(\varepsilon^{-2} \log^2 n)$$

leaves, this is done using the algorithm of Lemma 23 if the maximum matching is small compared to the number of disks in this subproblem, and the algorithm of Lemma 17 otherwise. The algorithm now returns the estimate returned by the algorithm of Lemma 6.

4.6.4 Analysis

► **Lemma 26.** *We have $\mathbb{E}[\sum_{v \in \mathcal{L}} m^*(\mathcal{D}_v)] \geq (1 - \varepsilon/4)m^*(\mathcal{D})$.*

► **Lemma 27.** *The running time of the above algorithm is $O(n \log n + \rho^9 \varepsilon^{-19} \log^2 n)$.*

► **Theorem 28.** *Given a set \mathcal{D} of n disks in the plane with density ρ , and a parameter $\varepsilon \in (0, 1)$, one can compute in $O(n \log n + \rho^9 \varepsilon^{-19} \log^2 n)$ time, a number Z , such that $(1 - \varepsilon)m^* \leq \mathbb{E}[Z]$ and $\mathbb{P}[Z > (1 + \varepsilon)m^*] < 1/n^{O(1)}$, where $m^* = m^*(\mathcal{D})$ is the size of the maximum matching in $\mathcal{I}_{\mathcal{D}}$.*

References

- 1 Boris Aronov and Sarel Har-Peled. On approximating the depth and related problems. *SIAM J. Comput.*, 38(3):899–921, 2008. doi:10.1137/060669474.
- 2 Paul Beame, Sarel Har-Peled, Sivaramakrishnan Natarajan Ramamoorthy, Cyrus Rashtchian, and Makrand Sinha. Edge estimation with independent set oracles. *ACM Trans. Algo.*, 16(4), September 2020. doi:10.1145/3404867.
- 3 Édouard Bonnet, Sergio Cabello, and Wolfgang Mulzer. Maximum matchings in geometric intersection graphs. In Christophe Paul and Markus Bläser, editors, *Proc. 37th Internat. Sympos. Theoret. Asp. Comp. Sci. (STACS)*, volume 154 of *LIPICs*, pages 31:1–31:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.STACS.2020.31.

- 4 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, Santa Clara, CA, USA, 3rd edition, 2008. doi:10.1007/978-3-540-77974-2.
- 5 Ran Duan and Seth Pettie. Linear-time approximation for maximum weight matching. *J. ACM*, 61(1), January 2014. doi:10.1145/2529989.
- 6 Alon Efrat, Alon Itai, and Matthew J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001. doi:10.1007/s00453-001-0016-8.
- 7 Harold N. Gabow and Robert Endre Tarjan. Faster scaling algorithms for general graph-matching problems. *J. Assoc. Comput. Mach.*, 38(4):815–853, 1991. doi:10.1145/115234.115366.
- 8 Sariel Har-Peled. A simple proof of the existence of a planar separator. *ArXiv e-prints*, April 2013. arXiv:1105.0103.
- 9 Sariel Har-Peled and Kent Quanrud. Approximation algorithms for polynomial-expansion and low-density graphs. *SIAM J. Comput.*, 46(6):1712–1744, 2017. doi:10.1137/16M1079336.
- 10 Sariel Har-Peled and Everett Yang. Approximation algorithms for maximum matchings in geometric intersection graphs. *CoRR*, abs/2201.01849, 2022. arXiv:2201.01849.
- 11 Nicholas J. A. Harvey. Algebraic algorithms for matching and matroid problems. *SIAM J. Comput.*, 39(2):679–702, 2009. doi:10.1137/070684008.
- 12 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973. doi:10.1137/0202019.
- 13 Haim Kaplan, Wolfgang Mulzer, Liam Roditty, Paul Seiferth, and Micha Sharir. Dynamic planar Voronoi diagrams for general distance functions and their algorithmic applications. *Discrete Comput. Geom.*, 64(3):838–904, September 2020. doi:10.1007/s00454-020-00243-7.
- 14 Zvi Lotker, Boaz Patt-Shamir, and Seth Pettie. Improved distributed approximate matching. *J. Assoc. Comput. Mach.*, 62(5):38:1–38:17, 2015. doi:10.1145/2786753.
- 15 Marcin Mucha and Piotr Sankowski. Maximum matchings via gaussian elimination. In *Proc. 45th Annu. IEEE Sympos. Found. Comput. Sci. (FOCS)*, pages 248–255. IEEE Computer Society, 2004. doi:10.1109/FOCS.2004.40.
- 16 Marcin Mucha and Piotr Sankowski. Maximum matchings in planar graphs via Gaussian elimination. *Algorithmica*, 45(1):3–20, 2006. doi:10.1007/s00453-005-1187-5.
- 17 Raphael Yuster and Uri Zwick. Maximum matching in graphs with an excluded minor. In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *Proc. 18th ACM-SIAM Sympos. Discrete Algs. (SODA)*, pages 108–117. SIAM, 2007. URL: <http://dl.acm.org/citation.cfm?id=1283383.1283396>.

The Complexity of the Hausdorff Distance

Paul Jungeblut  

Karlsruhe Institute of Technology, Germany

Linda Kleist  

Technische Universität Braunschweig, Germany

Tillmann Miltzow  

Utrecht University, The Netherlands

Abstract

We investigate the computational complexity of computing the Hausdorff distance. Specifically, we show that the decision problem of whether the Hausdorff distance of two semi-algebraic sets is bounded by a given threshold is complete for the complexity class $\forall\exists_{<}\mathbb{R}$. This implies that the problem is NP-, co-NP-, $\exists\mathbb{R}$ - and $\forall\mathbb{R}$ -hard.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Hausdorff Distance, Semi-Algebraic Set, Existential Theory of the Reals, Universal Existential Theory of the Reals, Complexity Theory

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.48

Related Version *Full Version:* <https://arxiv.org/abs/2112.04343>

Funding *Linda Kleist:* Partially supported by a postdoc fellowship of the German Academic Exchange Service (DAAD).

Tillmann Miltzow: Generously supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 016.Veni.192.250.

1 Introduction

The question of “how similar are two given objects” occurs in numerous settings. One typical tool to quantify their similarity is the Hausdorff distance. Two sets have a small Hausdorff distance if every point of one set is close to some point of the other set and vice versa. As a matter of fact, the Hausdorff distance appears in many branches of science. To illustrate the range of use cases, we consider two examples, for illustrations see Figure 1. In mathematics, the Hausdorff distance provides a metric on sets and henceforth also a topology. This topology can be used to discuss continuous transformations of one set to another [17]. In computer vision and geographical information science, the Hausdorff distance is used to measure the similarity between spacial objects [37, 45], for example the quality of quadrangulations of complex 3D models [52]. In this paper, we study the computational complexity of the Hausdorff distance from a theoretical perspective.

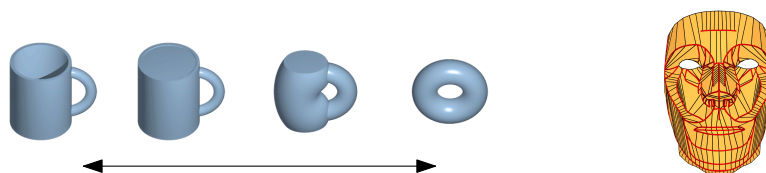
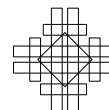
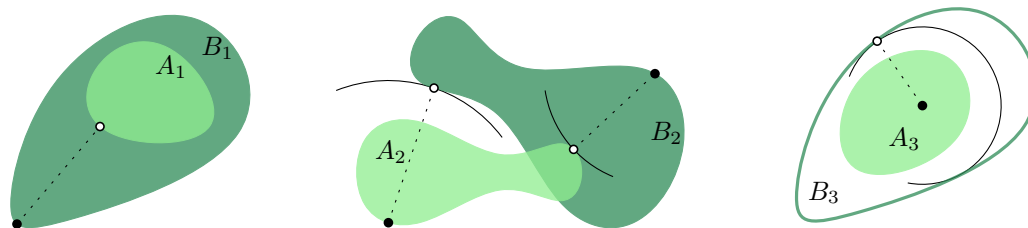


Figure 1 Left: Continuous deformation of a cup into a doughnut [22]. Right: Quadrangulation of a smooth surface used for rendering [52].



48:2 The Complexity of the Hausdorff Distance



■ **Figure 2** How similar are these sets?

Definition. The *directed Hausdorff distance* between a non-empty set $A \subseteq \mathbb{R}^n$ and a non-empty set $B \subseteq \mathbb{R}^n$ is defined as

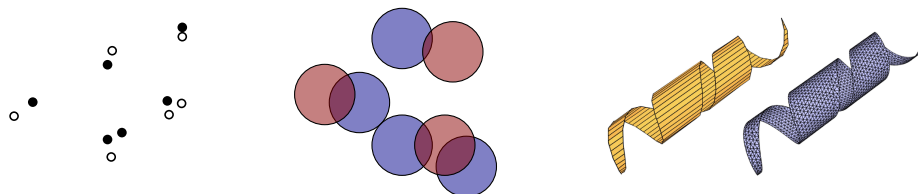
$$\vec{d}_H(A, B) := \sup_{a \in A} \inf_{b \in B} \|a - b\|.$$

The directed Hausdorff distance between A and B can be interpreted as the smallest value $\varepsilon \geq 0$ such that the (closed) ε -neighborhood of B contains A . Hence, it nicely captures the intuition of how much B has to be blown up to contain A . Note that $\vec{d}_H(A, B)$ and $\vec{d}_H(B, A)$ do not need to be equal, consider Figure 2: While $A \subset B$ and thus $\vec{d}_H(A, B) = 0$, it holds that $\vec{d}_H(B, A) > 0$. The (undirected) *Hausdorff distance* is symmetric and defined as

$$d_H(A, B) := \max\{\vec{d}_H(A, B), \vec{d}_H(B, A)\}.$$

In this paper, we investigate the *computational complexity* of deciding whether the Hausdorff distance of two sets is at most a given threshold.

Semi-algebraic sets. The algorithmic complexity of computing the Hausdorff distance clearly depends on the type of their underlying sets. If we are given the sets in a way that we cannot even decide if they are empty, it seems near impossible to compute their Hausdorff distance. However, if the sets consist of finitely many points, their Hausdorff distance can easily be computed by checking all pairs of points. In practice, we are often somewhere between those two extreme situations. For instance, the sets could be a collection of disks in the plane or cubic splines, describing a surface in three dimensions, see also Figure 3.



■ **Figure 3** The Hausdorff distance can appear in simpler or more complicated settings. Left: Two finite point sets (black and white) in the plane. Middle: Two sets of blue and red disks in the plane. Right: Two surfaces in 3-space with different meshes, image taken from [52].

In this paper, we focus on semi-algebraic sets defined over the ring of integers, i.e., sets that can be described by polynomial inequalities with integer coefficients. For simplicity, we just write *semi-algebraic set*, and silently assume all coefficients of defining polynomials are integers. Formally, a semi-algebraic set is the finite union of basic semi-algebraic sets. A

basic semi-algebraic set S is specified by two families of polynomials \mathcal{P} and \mathcal{Q} with integer coefficients such that

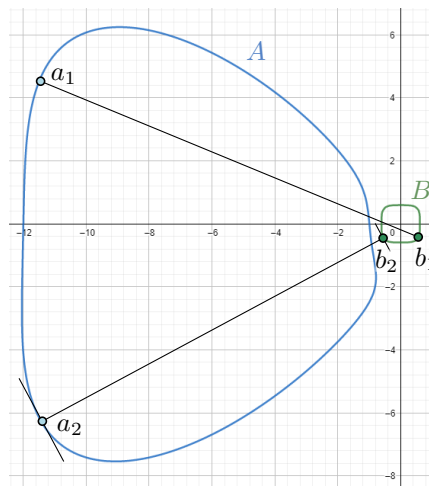
$$S = \left\{ x \in \mathbb{R}^n \mid \bigwedge_{P \in \mathcal{P}} P(x) \leq 0 \wedge \bigwedge_{Q \in \mathcal{Q}} Q(x) < 0 \right\}.$$

Semi-algebraic sets cover clearly the vast majority of practical cases. Simultaneously, one deals with polynomials even in supposedly simple cases, i.e., when considering cubic splines.

Concrete example. The following example was made up on the spot by Bernd Sturmfels at a workshop in Saarbrücken in 2019. The two polynomials

$$\begin{aligned} f(x, y) &:= x^4 + y^4 + 12x^3 + 2y^3 - 3xy + 11 \\ g(x, y) &:= 7x^4 + 8y^4 - 1 \end{aligned}$$

define the sets $A = \{(x, y) \in \mathbb{R}^2 \mid f(x, y) = 0\}$ and $B = \{(x, y) \in \mathbb{R}^2 \mid g(x, y) = 0\}$. For an illustration of A and B , consider the blue and green curve in Figure 4, respectively.



■ **Figure 4** The Hausdorff distance between the compact semi-algebraic sets (in blue and green) is attained at points (a_2, b_2) such that the segment a_2b_2 is orthogonal to the tangents at a_2 and b_2 . While the segment a_1b_1 is longer than a_2b_2 , the pair (a_1, b_1) does not realize the Hausdorff distance because the segment a_1b_1 crosses both A and B .

It can be argued using convexity and continuity that the Hausdorff distance is attained at points $a \in A, b \in B$ such that the segment ab is orthogonal to the tangents at a and b . This yields a set of polynomial equations in four variables. The system has 240 complex solutions, eight of which are real. These 240 solutions can be computed using computer algebra systems based on Gröbner bases. For some of the real solutions (a, b) , the segment ab crosses A and B , for example a_1b_1 as in Figure 4. Among the remaining solutions the points $a_2 \approx (-11.48362, -6.1760), b_2 \approx (-0.56460, -0.43583)$ realize the Hausdorff distance of approximately 12.33591. This approach does not easily generalize to general semi-algebraic sets. In the next paragraph, we present a slower, but more general method.

General decision algorithm. We consider a situation where we are given two semi-algebraic sets A and B as well as a threshold t ; for simplicity, we assume here (only in this paragraph) that A and B are closed. The statement $\vec{d}_H(A, B) \leq t$ can be encoded into a logical sentence

Φ of the form $\forall a \in A. \exists b \in B : \|a - b\|^2 \leq t^2$, where $\|x\|$ denotes the Euclidean norm of the vector x . We can decide the truth of this sentence by employing sophisticated algorithms from real algebraic geometry that can deal with *two blocks of quantifiers* [12, Chapter 14]. These algorithms are so slow that they would probably not work in the above example. Our main result roughly states that in general there is little hope for an improvement. To state this formally, we continue by defining suitable complexity classes.

Algorithmic complexity. Let φ be a *quantifier-free formula in the first-order theory of the reals*, i.e., a formula formed over the alphabet $\Sigma = \{\mathbb{Z}, +, \cdot, =, \leq, <, \vee, \wedge, \neg\}$ together with symbols for the variables. Details on how formulas are encoded are described in Section 2. The UNIVERSAL EXISTENTIAL THEORY OF THE REALS (UETR) asks to decide the truth value of a sentence

$$\Phi := \forall X \in \mathbb{R}^n . \exists Y \in \mathbb{R}^m : \varphi(X, Y).$$

An instance of UETR belongs to STRICT-UETR if the corresponding formula φ is over the alphabet $\Sigma = \{\mathbb{Z}, +, \cdot, <, \vee, \wedge\}$, i.e., if every atom is a strict inequality and negations do not occur. The complexity classes $\forall\exists\mathbb{R}$ and $\forall\exists_{<}\mathbb{R}$ contain all decision problems for which there exists a polynomial-time many-one reduction to UETR and STRICT-UETR, respectively. We propose to pronounce the complexity class $\forall\exists\mathbb{R}$ as “UER” or “forall exists R” and $\forall\exists_{<}\mathbb{R}$ as “Strict-UER” or “strict forall exists R”. Let us emphasize that we work in the bit-model of computation; all inputs have finite precision and their overall length determines the size of the problem instance. To the best of our knowledge, $\forall\exists\mathbb{R}$ was first introduced by Bürgisser and Cucker [19, Section 9] under the name $\text{BP}^0(\forall\exists)$ (in the constant-free Blum-Shub-Smale-model [16]). The notation $\forall\exists\mathbb{R}$ arised later in [27] extending the notation from Schaefer and Števanekovič [48]. The class $\text{co-}\forall\exists_{<}\mathbb{R} = \exists\forall_{\leq}\mathbb{R}$ was first studied by D’Costa, Lefauchaux, Neumann, Ouaknine and Worrel [25].

Concerning the relation of these complexity classes, it is easy to see that $\forall\exists_{<}\mathbb{R}$ is contained in $\forall\exists\mathbb{R}$. It is an intriguing open problem if those two classes coincide or are different. If the two classes are indeed different, this would imply $\text{NP} \neq \text{PSPACE}$ so we do not expect such a proof any time soon. It is also conceivable that some extensions of known results in real algebraic geometry can be used to show $\forall\exists\mathbb{R} = \forall\exists_{<}\mathbb{R}$.

Problem and results. We now have all ingredients to state our problem and main results. Let $\Phi_A(X)$ and $\Phi_B(X)$ be two quantifier-free formulas defining the semi-algebraic sets $A = \{x \in \mathbb{R}^n \mid \Phi_A(x)\}$ and $B = \{x \in \mathbb{R}^n \mid \Phi_B(x)\}$, and let $t \in \mathbb{Q}$ be a rational number. The HAUSDORFF problem asks whether $d_H(A, B) \leq t$. Here the dimension n of the ambient space of A and B is part of the input (there is a polynomial-time algorithm for every fixed n , see the related work in Section 1.1). Our main result determines the algorithmic complexity.

► **Theorem 1.** *The HAUSDORFF problem is $\forall\exists_{<}\mathbb{R}$ -complete.*

Note that prior to our result, it was not even known if computing the Hausdorff distance was NP-hard. As $\forall\exists_{<}\mathbb{R}$ contains NP, co-NP, $\exists\mathbb{R}$ and $\forall\mathbb{R}$, we also get hardness for all of those complexity classes. Theorem 1 answers an open question posed by Dobbins, Kleist, Miltzow and Rzażewski [27].

One may wonder whether it is crucial for our results that the HAUSDORFF problem asks for the distance to be *at most* t rather than *below* t . We remark that all our proofs work with tiny modifications also for the case of a strict inequality. Furthermore, our results also hold for the directed Hausdorff distance. Note that one can compute the undirected Hausdorff

distance trivially, by computing twice the directed Hausdorff distance. Thus intuitively, the directed Hausdorff distance is computationally at least as hard. Yet, this is not a many-one reduction, as we need to compute the directed Hausdorff distance twice.

In the proof of $\forall\exists_{<}\mathbb{R}$ -hardness for Theorem 1, we create instances with some additional properties. In particular, we can guarantee a gap, i.e., the Hausdorff distance is either below the threshold t or at least $t \cdot 2^{2^{\Omega(d)}}$, where d denotes the number of variables of Φ_A and Φ_B . Thus our result also rules out approximation algorithms.

► **Corollary 2.** *Let A and B be two semi-algebraic sets in \mathbb{R}^d and $f(d) = 2^{2^{o(d)}}$. Then there is no polynomial-time $f(d)$ -approximation algorithm to compute $d_{\text{H}}(A, B)$, unless $\text{P} = \forall\exists_{<}\mathbb{R}$.*

We remark that our proof provides hard instances, where the threshold t is strictly larger than zero. By scaling of A and B , we can assume $t = 1$ without loss of generality. It is natural to wonder if $\forall\exists_{<}\mathbb{R}$ -hardness also holds for the case of $t = 0$. This question is equivalent to checking whether the closure of two semi-algebraic sets is equal, i.e., $d_{\text{H}}(A, B) = 0$ if and only if $\overline{A} = \overline{B}$. Computing the closure of a semi-algebraic set is non-trivial. In particular, it is not enough to replace all occurrences of $<$ by \leq . Yet testing, if two semi-algebraic sets are equal is likely slightly easier.

► **Theorem 3.** *Deciding if two semi-algebraic sets are equal is $\forall\mathbb{R}$ -complete.*

Because the proof is rather simple, we present it at this point.

Proof. Given quantifier-free formulas $\Phi_A(X)$ and $\Phi_B(X)$, it holds that $A = B$ if and only if $\forall X \in \mathbb{R}^n : \Phi_A(X) \iff \Phi_B(X)$. This shows $\forall\mathbb{R}$ -membership. To see $\forall\mathbb{R}$ -hardness, note that $\Psi := \forall X \in \mathbb{R}^n : \varphi(X)$ is equivalent to $\{x \in \mathbb{R}^n : \varphi(x)\} = \mathbb{R}^n$. ◀

1.1 Related work

This subsection reviews previous work concerning two directions. First, we discuss the complexity of computing the Hausdorff distance for special sets. Afterwards, we investigate previous work on the complexity class $\forall\exists\mathbb{R}$.

Computing the Hausdorff distance. The notion of the Hausdorff distance was introduced by Felix Hausdorff in 1914 [32]. Most of the early works focused on the Hausdorff distance for finite point sets. For a set of n points and a set of m points in any fixed dimension, the Hausdorff distance can be easily computed by checking all pairs, i.e., in time $O(mn)$. In the plane, this can be improved to $O((n+m)\log(m+n))$ by using Voronoi diagrams [7]. In fact, this method can be extended to sets consisting of pairwise non-crossing line segments in the plane, e.g., simple polygons and polygonal chains fulfill this property. If the polygons are additionally convex, their Hausdorff distance can even be computed in linear time [11].

More generally, the Hausdorff distance can be computed in polynomial time whenever the two sets can be described by a simplicial complex of fixed dimension. Based on the PhD thesis of Godau [30], Alt et al. [8, Theorem 3.3] show how to compute the directed Hausdorff distance between two sets in \mathbb{R}^d consisting of n and m k -dimensional simplices in time $O(nm^{k+2})$ (assuming d is constant). Using a Las Vegas algorithm for computing the vertices of the lower envelope, similar ideas yield an approach with randomized expected time in $O(nm^{k+\varepsilon})$ for $k > 1$ and every $\varepsilon > 0$ [8, Theorem 3.4]. They additionally present algorithms with better randomized expected running times for sets of triangles in \mathbb{R}^3 and point sets in \mathbb{R}^d .

Given two semi-algebraic sets $A, B \subseteq \mathbb{R}^n$, the HAUSDORFF problem can be encoded as a sentence of the form $\forall X \exists Y : \varphi(X, Y)$ with $\Theta(n)$ variables, where φ is quantifier-free. Such a sentence can be decided in time roughly equal to $(sd)^{O(n^2)}$ [12, Theorem 14.14] where d denotes the maximum degree of any polynomial of φ and s denotes the number of atoms.

In other contexts the two sets are allowed to undergo certain transformations (e.g. translations) such that the Hausdorff distance is minimized [18]. See Alt [9] for a survey.

Universal existential theory of the reals. As mentioned above, the complexity class $\forall\exists\mathbb{R}$ was first studied by Bürgisser and Cucker who prove complexity results for many decision problems involving circuits [19]. For example, they study functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that are given by arithmetic circuits. They show that it is $\forall\exists\mathbb{R}$ -complete to decide if such f is surjective. Dobbins, Kleist, Miltzow, and Rzażewski [28, 27] consider $\forall\exists\mathbb{R}$ in the context of area-universality of graphs. A plane graph is *area-universal* if for every assignment of reals to the inner faces of a plane graph, there exists a straight-line drawing such that the area of each inner face equals the assigned number. Dobbins et al. conjecture that the decision problem whether a given plane graph is area-universal is complete for $\forall\exists\mathbb{R}$. They support this conjecture by proving hardness for several related notions [27]. Additionally, for future research directions, they present a number of candidates for potentially $\forall\exists\mathbb{R}$ -hard problems. Among them, they stated a question motivating this paper as an open problem, namely whether the HAUSDORFF problem is $\forall\exists\mathbb{R}$ -complete. The other candidates exhibit intrinsic connections to the notions of imprecision, robustness and extendability.

We point out that the computational complexity may also become easier when asking universal-type questions. For example, it is $\exists\mathbb{R}$ -complete to decide whether a graph is a unit distance graph, i.e., whether it has a straight-line drawing in the plane in which all edges have the same length [47]. On the other hand, the decision problem whether for all reasonable assignments of weights to the edges, a graph has a straight-line drawing in which the edge lengths correspond to the assigned weight lies in P [14]. Similarly, it is $\exists\mathbb{R}$ -complete to decide for a given planar graph for which some vertices are fixed to the boundary of a polygon (with holes) whether there exists a planar straight-line drawing inside the polygon [33]. The case of simple polygons is open. In contrast, there is a polynomial time algorithm to test if a given graph G and a contained cycle C admit for *every* simple polygon P , representing C , a straight-line drawing of G inside P [39].

The sister class $\exists\forall\mathbb{R}$ was recently investigated by D’Costa et al. [25]. They show that it is $\exists\forall_{\leq}\mathbb{R}$ -complete to decide for a given rational matrix A and a compact semi-algebraic set $K \subseteq \mathbb{R}^n$, whether there exists a starting point $x \in K$ such that $x_n := A^n x$ is contained in K for all $n \in \mathbb{N}$. This and similar problems are generally referred to as *escape* problems.

The complexity class $\forall\exists\mathbb{R}$ is a natural extension of the complexity class $\exists\mathbb{R}$ (pronounced as “exists \mathbb{R} ”, “ER”, or “ETR”), which is defined similarly to $\forall\exists\mathbb{R}$, but without universally quantified variables. The complexity class $\exists\mathbb{R}$ has gained a lot of interest in recent years, specifically in the computational geometry community. It gains its significance because numerous well-studied problems from diverse areas of theoretical computer science and mathematics have been shown to be complete for this class. Famous examples from discrete geometry are the recognition of geometric structures, such as unit disk graphs [35], segment intersection graphs [34], visibility graphs [21], stretchability of pseudoline arrangements [38, 50], and order type realizability [34]. Other $\exists\mathbb{R}$ -complete problems are related to graph drawing [33], Nash-Equilibria [15, 29], geometric packing [6], the art gallery problem [3], convex covers [2], non-negative matrix factorization [49], polytopes [26, 43], geometric embeddings of simplicial complexes [4], geometric linkage constructions [1], training neural

networks [5], and continuous constraint satisfaction problems [36]. For more information on the complexity class $\exists\mathbb{R}$, we refer to Matoušek’s lecture notes [34], and the surveys by Schaefer [46] and Cardinal [20].

General solution strategies. We sometimes see that researchers make the *dichotomy* between tractable and intractable algorithmic problems. More precisely, when there exists a polynomial time algorithm the underlying problem is considered to be tractable. In contrast, in case of NP-hardness the underlying problem is considered intractable. Although most researchers are aware that this dichotomy does not match actual practical performance, it is often seen as a good enough yardstick.

In the last decades, a more *nuanced* perspective emerged. This new perspective acknowledges that there is a whole range of mathematical assumptions and models and that depending on the specific situation, different models can be more or less accurate [44]. One example is the so-called *smoothed analysis* of algorithms [51]. The underlying idea is that practical instances are subject to small noise. This small noise may tame a very difficult instance. In this context, we discuss four complexity classes: NP, $\exists\mathbb{R}$, Π_2^p , and $\forall\exists\mathbb{R}$.

- NP Despite NP-hardness, huge practical instances can often be solved very fast. Prominent examples are ILPs that can be solved optimally using off the shelf solvers. Note that it is also possible to generate adversarial instances of moderate size for which no good tools exist.
- $\exists\mathbb{R}$ Problems in $\exists\mathbb{R}$ are considerably harder. Still, we can often solve $\exists\mathbb{R}$ -complete problems using suitable discretizations or using gradient descent. However, both methods usually have no guarantees to ever terminate. Furthermore, they may give solutions that are arbitrarily far from the optimum. Methods from real algebraic geometry are applicable if polynomials are explicitly given and contain only few variables, say around ten.
- Π_2^p Describes problems on the second level of the polynomial time hierarchy [10]. We do not know many problems on this level, compared to the number of NP-complete problems. Due to the two blocks of quantifiers there are no effective general purpose tools like ILP-solvers. On the positive side, due to the combinatorial nature, it is possible to use exhaustive search.
- $\forall\exists\mathbb{R}$ This class combines the difficulties of $\exists\mathbb{R}$ and Π_2^p . Note that we cannot even use gradient descent for problems in this class. Due to the continuous nature of the problem it is also not possible to use a simple brute-force algorithm. Furthermore, methods from real algebraic geometry cannot even solve small instances with up to say ten variables. The two different quantifiers limit those already impractical methods even further.

We want to point out that this classification of difficulty should not be taken dogmatically. For many algorithmic problems worst-case complexity is not an adequate model to explain practical performance. We rather take the perspective that this mathematical classification is a crude yardstick which measures algorithmic difficulty from the worst-case perspective. For each individual problem one has to judge, if the worst-case perspective is accurate.

1.2 Techniques and proof overview

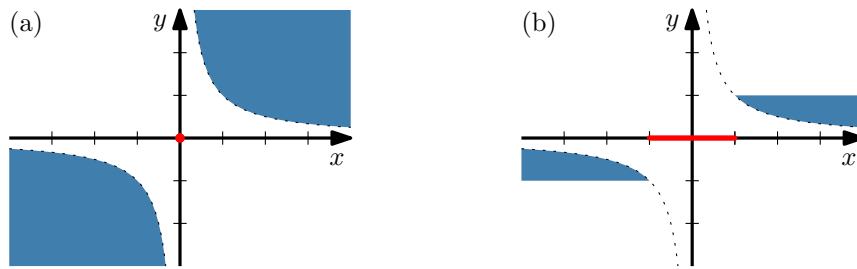
In this subsection, we present the general idea behind the hardness reduction for the HAUSDORFF problem. The goal is to convey the intuition and to motivate the technical intermediate steps needed. The sketched reduction is oversimplified and thus neither in polynomial time nor fully correct. We point out both of these issues and give first ideas on how to solve them.

Let $\Phi := \forall X \in \mathbb{R}^n . \exists Y \in \mathbb{R}^m : \varphi(X, Y)$ be a STRICT-UETR instance. We define two sets

$$A := \{x \in \mathbb{R}^n \mid \exists Y \in \mathbb{R}^m : \varphi(x, Y)\} \quad \text{and} \quad B := \mathbb{R}^n$$

and ask whether $d_H(A, B) = 0$. If Φ is true, then $A = \mathbb{R}^n$ and we have $d_H(A, B) = 0$ because both sets are equal. Otherwise, if Φ is false, then there exists some $x \in \mathbb{R}^n$ for which there is no $y \in \mathbb{R}^m$ satisfying $\varphi(x, y)$ and we conclude that $A \neq \mathbb{R}^n$. In general we call the set of all $x \in \mathbb{R}^n$ for which there is no $y \in \mathbb{R}^m$ satisfying $\varphi(x, y)$ the *counterexamples* $\perp(\Phi)$ of Φ . One might hope that $\perp(\Phi) \neq \emptyset$ is enough to obtain $d_H(A, B) > 0$, but this is not the case. To this end, consider the formula $\Psi := \forall X \in \mathbb{R}. \exists Y \in \mathbb{R} : XY > 1$, which is false. The set $\perp(\Psi) = \{0\}$ contains only a single element, so we have $A = \mathbb{R} \setminus \{0\}$ and $B = \mathbb{R}$. However, their Hausdorff distance also evaluates to $d_H(A, B) = 0$. We conclude that above reduction does not (yet completely) work, because it maps a yes- and a no-instance of STRICT-UETR to a yes-instance of HAUSDORFF.

We solve this issue by blowing up the set of counter examples. Specifically, Theorem 10 establishes a polynomial-time algorithm to transform a STRICT-UETR instance Φ into an equivalent formula Φ' such that the set of counterexamples is either empty (if Φ' is true) or contains an open ball of positive radius (if Φ' is false). The radius of the ball serves as a lower bound on the Hausdorff distance $d_H(A, B)$. Thus a reduction starting with Φ' is correct. As a key tool for this step, we restrict the variable ranges from \mathbb{R}^n and \mathbb{R}^m to small and compact intervals. Figure 5 presents an example on how such a range restriction may enlarge the set of counterexamples from a single point to an interval.



■ **Figure 5** Consider the formula $\forall X \in \mathbb{R}. \exists Y \in \mathbb{R} : XY > 1$. (a) Each point $(x, y) \in \mathbb{R}^2$ in the blue open region satisfies $xy > 1$. Only for $x = 0$ (in red) no suitable $y \in \mathbb{R}$ exists. (b) Restricting the range of Y to $[-1, 1]$, then for all $x \in [-1, 1]$ (in red) no y with $xy > 1$ exists.

We highlight that such a restriction of the variable ranges is not possible for general UETR formulas. However, we can exploit the fact that STRICT-UETR formulas are \forall -strict; a negation- and implication-free formula is \forall -strict if each atom involving universally quantified variables is a strict inequality. Being \forall -strict is a key property of many of the formulas considered throughout the paper, both for $\forall \exists < \mathbb{R}$ -hardness and -membership. We think that the special property of blown up counterexamples can prove useful in future reductions to show $\forall \exists < \mathbb{R}$ -hardness of other problems because it makes handling the no-instances easier.

A further challenge is given by the definition of the sets A and B . While the description complexity of B depends only on n , the definition of A contains an existential quantifier. This is troublesome because our definition of the HAUSDORFF problem requires quantifier-free formulas as its input, and in general there is no equivalent quantifier-free formula of polynomial length which describes the set A [24]. We overcome this issue by taking the existentially quantified variables as additional dimensions into account; it will be useful to scale them to a tiny range, so that their influence on the Hausdorff distance becomes negligible. Therefore instead of the above, in Section 5 we work with sets similar to

$$A := \{(x, y) \mid x \in [-1, 1]^n, y \in [-\varepsilon, \varepsilon]^m, \varphi(x, y)\} \quad \text{and} \\ B := [-1, 1]^n \times \{0\}^m$$

for some tiny value ε depending on the radius r (of the ball contained in the counterexamples) computed in Section 4. This definition of A and B introduces the new issue that even if Φ is true, the Hausdorff distance $d_H(A, B)$ might be strictly positive. However, we manage to identify a threshold t , such that $d_H(A, B) \leq t$ if and only if Φ is true. This completes the proof of $\forall\exists_{<} \mathbb{R}$ -hardness.

Organization. The remainder of the paper is organized as follows. We introduce preliminaries concerning the first-order theory of the reals in Section 2 and essential tools from real algebraic geometry in Section 3. Section 4 presents the result for blowing up the set of counterexamples for \forall -strict formulas and Section 5 the hardness proof. For the membership of HAUSDORFF in $\forall\exists_{<} \mathbb{R}$ we refer to Theorem 17 of the full version. We conclude with a list of open problems in Section 6. Statements marked with (\spadesuit) are proved in the full version.

2 Preliminaries on the first-order theory of the reals and $\forall\exists\mathbb{R}$

Here, we give a short overview of the notation and definitions used in the paper. We mostly introduce standard terminology following the book by Cox, Little, O’Shea [23].

An *atom* is an expression of the form $P \circ 0$ for some polynomial $P \in \mathbb{Z}[X_1, \dots, X_n]$ and $\circ \in \{<, \leq, =, \neq, \geq, >\}$. We always assume that a polynomial is written as a sum of monomials. Its *total degree* is the maximum number of occurrences of variables involved in any monomial. For example $P(X, Y, Z) = X^2Y^2 + XYZ$ has total degree four. A variable is called *free* if it is not bound by a quantifier. A *formula* is either (i) an atom, or (ii) if φ_1, φ_2 are formulas, then $\varphi_1 \wedge \varphi_2, \varphi_1 \vee \varphi_2, \varphi_1 \implies \varphi_2$ and $\neg\varphi_1$ are formulas, or (iii) if X is a free variable of a formula $\varphi(X)$, then $\exists X : \varphi(X)$ and $\forall X : \varphi(X)$ are formulas in which X is bound. In order to determine the *length* $|\varphi|$ of a formula φ , we count 1 for each fixed symbol, we encode integer coefficients in binary, exponents are encoded in unary, and we count $\log n$ for every occurrence of each variable, where n denotes the number of variables. We denote by QFF the family of quantifier-free formulas that contain no negation or implication. Furthermore, $\text{QFF}_{<}, \text{QFF}_{\leq},$ and $\text{QFF}_{=}$ are the families in QFF that have only atoms involving $<, \leq$ and $=$ respectively.

A *sentence* is a formula without free variables and thus either equivalent to true or to false. The truth value is defined inductively, by interpreting the quantifiers over the real numbers \mathbb{R} . As a convention, we use capitalized Greek letters for sentences and use lower case Greek letter for formulas. We write $\Psi \equiv \Psi'$ if the two sentences have the same truth value. The *first order theory of the reals* (FOTR) is the family of all true sentences. If all quantifiers of a formula appear at its beginning, we say it is in *prenex normal form*. We usually write *blocks of variables*, i.e., $\forall X \in \mathbb{R}^n : \varphi(X)$. Here X is a shorthand notation for $X = (X_1, \dots, X_n)$. We say n is the length of X in this case. All quantifiers quantify their bound variables over \mathbb{R} . The following are just shorthand notation:

$$\begin{aligned} \forall X \in [-1, 1] : \varphi(X) &\equiv \forall X \in \mathbb{R} : (X \geq -1 \wedge X \leq 1) \implies \varphi(X) \\ \exists X \in [-1, 1] : \varphi(X) &\equiv \exists X \in \mathbb{R} : (X \geq -1 \wedge X \leq 1) \wedge \varphi(X) \end{aligned}$$

We use uppercase letters for variables in formulas and lowercase letters for specific values, i.e., symbol X denotes a vector of variables, while $x \in \mathbb{R}^n$ is a point. We sometimes write $\varphi(X, Y)$ to emphasize that X and Y are free variables of the formula φ . Often we do not mention the free variables of φ though.

48:10 The Complexity of the Hausdorff Distance

Consider a formula $\Phi := \forall X \in \mathbb{R}^n . \exists Y \in \mathbb{R}^m : \varphi(X, Y)$, where $\varphi \in \text{QFF}$. Each atom of φ is of the form $P \circ 0$, where $\circ \in \{<, \leq, =, \neq, \geq, >\}$ and $P \in \mathbb{Z}[X, Y]$ is a multivariate polynomial in the variables X and Y . Without loss of generality we can restrict our attention to the case of $\circ \in \{<, \leq\}$, because the following transformations show that the other relations can be reformulated such that the length of the formula is at most doubled.

$$\begin{aligned} P > 0 &\equiv -P < 0 & P = 0 &\equiv (P \leq 0) \wedge (-P \leq 0) \\ P \geq 0 &\equiv -P \leq 0 & P \neq 0 &\equiv (P < 0) \vee (-P < 0) \end{aligned}$$

Furthermore, we can assume that φ contains only the logical connectives \wedge and \vee , because De Morgan's law allows to push all negations (and therefore also implications) down to the atoms transforming φ into *negation normal form*. With the following equivalences we obtain a formula without negations:

$$\neg(P < 0) \equiv -P \leq 0 \qquad \neg(P \leq 0) \equiv -P < 0$$

Given a formula φ , the set $S(\varphi) = \{x \in \mathbb{R}^n \mid \varphi(x)\}$ is semi-algebraic. The *complexity* of a semi-algebraic set S is the length of a shortest quantifier-free formula φ , such that $S = S(\varphi)$ (recall that integers are encoded in binary). We write $\varphi \equiv \varphi'$ if $S(\varphi) = S(\varphi')$.

For any fixed $\circ \in \{<, \leq\}$, we denote by $\forall \exists \circ \mathbb{R}$ the fragment of $\forall \exists \mathbb{R}$ containing all decision problems that polynomial-time many-one reduce to a UETR-instance where all formulas are contained in QFF_\circ . Similarly, for $\circ \in \{<, \leq\}$, we denote the corresponding fragments of $\exists \mathbb{R}$ and $\forall \mathbb{R}$ by $\exists \circ \mathbb{R}$ and $\forall \circ \mathbb{R}$, respectively. The following lemma summarizes what we know about the relation between the complexity classes $\forall \exists < \mathbb{R}$, $\forall \exists \leq \mathbb{R}$ and $\forall \exists \mathbb{R}$ as well as their relation to the well-studied classes NP, co-NP, $\exists \mathbb{R}$, $\forall \mathbb{R}$, and PSPACE.

► **Lemma 4 (♠).** *It holds $\text{NP} \subseteq \exists \mathbb{R} \subseteq \forall \exists < \mathbb{R} \subseteq \forall \exists \leq \mathbb{R} = \forall \exists \mathbb{R} \subseteq \text{PSPACE}$. Furthermore, $\text{co-NP} \subseteq \forall \mathbb{R} \subseteq \forall \exists < \mathbb{R}$.*

3 Mathematical tools

In this section, we review already existing tools that are needed throughout the paper. In particular, we use two sophisticated results from algebraic geometry, namely singly exponential quantifier elimination and the so called Ball Theorem. While quantifier elimination provides equivalent quantifier free formulas of bounded length, the Ball Theorem guarantees that every non-empty semi-algebraic set contains an element not too far from the origin. We use the two results to establish useful properties of semi-algebraic sets.

We start with a result on quantifier-elimination which originates from a series of articles by Renegar [40, 41, 42]. We note that the time complexity of this algorithm is exponential and not doubly exponential for every fixed number of quantifier alternations.

► **Theorem 5** ([12, Theorem 14.16]). *Let X_1, \dots, X_k, Y be vectors of real variables where X_i has length n_i , Y has length m , formula $\varphi(X_1, \dots, X_k, Y) \in \text{QFF}$ has s atoms and $Q_i \in \{\exists, \forall\}$ is a quantifier for all $i = 1, \dots, k$. Further, let d be the maximum total degree of any polynomial of $\varphi(X_1, \dots, X_k, Y)$. Then for any formula $\Phi(Y) := (Q_1 X_1) \dots (Q_k X_k) : \varphi(X_1, \dots, X_k, Y)$ there is an equivalent quantifier-free formula of size at most*

$$s^{(n_1+1) \dots (n_k+1)(m+1)} d^{O(n_1) \dots O(n_k)O(m)}.$$

We use the following corollary of Theorem 5 that is weaker but easier to work with.

► **Corollary 6** (♠). *Given a formula $\Phi(Y)$ as in Theorem 5 of length $L = |\varphi(X_1, \dots, X_n, Y)|$. Then for a constant $\alpha \in \mathbb{R}$ independent of Φ , there exists an equivalent quantifier-free formula of size at most $L^{\alpha^{k+1} \cdot n_1 \cdot \dots \cdot n_k \cdot m}$.*

The Ball Theorem was first discovered by Vorob'ev [53] and Grigor'ev and Vorobjov [31]. Vorob'ev and Vorobjov are two different transcriptions of the same name from the Cyrillic to the Latin alphabet. Explicit bounds on the distance are given by Basu and Roy [13]. We use a formulation from Schaefer and Štefankovič [48].

► **Theorem 7** (Ball Theorem [48, Corollary 3.1]). *Every non-empty semi-algebraic set in \mathbb{R}^n of complexity at most $L \geq 4$ contains a point of distance at most $2^{L^{8n}}$ from the origin.*

Recall that for any quantifier-free formula $\varphi(X)$ with free variables $X \in \mathbb{R}^n$, the set $S := \{x \in \mathbb{R}^n \mid \varphi(x)\}$ is semi-algebraic. Thus, a direct conclusion of Theorem 7 is that $\exists X \in \mathbb{R}^n : \varphi(X)$ is equivalent to $\exists X \in [-2^{L^{8n}}, 2^{L^{8n}}]^n : \varphi(X)$. This is how we are going to make use of Theorem 7 throughout this paper.

In the following, we deduce useful properties from Corollary 6 and Theorem 7, starting with a fact that was identified by D'Costa, Lefauchaux, Neumann, Ouaknine and Worrel [25, Lemma 14] for two quantifiers. We are interested in a generalization to more quantifiers. Their proof also works with slight modifications in the more general case with k quantifiers.

► **Lemma 8** (♠). *Let X_1, \dots, X_k be vectors of variables where X_i has length $n_i \geq 1$ and let $\varphi(\varepsilon, X_1, \dots, X_k)$ be a quantifier-free formula of length L . Then the semi-algebraic set*

$$S = \{\varepsilon > 0 \mid (Q_1 X_1) \dots (Q_k X_k) : \varphi(\varepsilon, X_1, \dots, X_k)\},$$

where the Q_i are alternating existential and universal quantifiers, is either empty or it contains an element $\varepsilon^* \in S$ such that for some constant $\beta \in \mathbb{R}$ we have $\varepsilon^* \geq 2^{-L^{\beta^{k+2} \cdot n_1 \cdot \dots \cdot n_k}}$.

Given a semi-algebraic set $S \subseteq \mathbb{R}^n$ and any $\alpha \in \mathbb{Q}$, the scaled set $T = \{\alpha x \in \mathbb{R}^n \mid x \in S\}$ is semi-algebraic. The following lemma proves that scaling any subset of the variables by a doubly exponentially large integer can be encoded by a formula of polynomial length.

We denote by the *type* of an atom whether it is a strict inequality, a non-strict inequality or an equation. We say that two formulas *have the same logical structure* if there is a bijection between their atoms such that identifying corresponding atoms leads to the same formula.

► **Lemma 9** (Scaling Semi-Algebraic Sets ♠). *Let $\varphi(X, Y) \in \text{QFF}$ with free variables $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}^m$. Further, let N be an integer and $s \in \{-1, 1\}$. We can construct in time polynomial in $|\varphi|$ and N a formula $\psi(X, Y)$, such that for any $(x, y) \in \mathbb{R}^{n+m}$ we have $\varphi(x, y)$ if and only if $\psi(x \cdot 2^{s \cdot 2^N}, y)$. Further $\psi(X, Y)$ can be chosen to be of the form*

$$\begin{aligned} \psi(X, Y) &\equiv \exists U \in [-1, 1]^{N+1} : \chi(U) \wedge \varphi'(X, Y, U) \quad \text{or alternatively} \\ \psi(X, Y) &\equiv \forall U \in [-1, 1]^{N+1} : \neg \chi(U) \vee \varphi'(X, Y, U). \end{aligned}$$

In both cases, $\chi(U) \in \text{QFF}_{=}$, formulas $\varphi'(X, Y, U)$ and $\varphi(X, Y)$ have the same logical structure and corresponding atoms have the same type.

4 Counterexamples of Strict-UETR

Let us recall the definition of *counterexamples* here that was already motivated in Section 1.2. Given a sentence $\Phi := \forall X \in \mathbb{R}^n . \exists Y \in \mathbb{R}^m : \varphi(X, Y)$ we call the set

$$\perp(\Phi) := \{x \in \mathbb{R}^n \mid \forall Y \in \mathbb{R}^m : \neg \varphi(x, Y)\}$$

48:12 The Complexity of the Hausdorff Distance

its *counterexamples*. The counterexamples of Φ are exactly the values $x \in \mathbb{R}^n$ for which there is no $y \in \mathbb{R}^m$ such that $\varphi(x, y)$ is true. We show how to transform a STRICT-UETR instance Φ into an equivalent formula Ψ for which $\perp(\Psi)$ is either empty or contains an open ball. We achieve this by bounding the range over which the variables are quantified. The following theorem summarizes our findings. This open ball property is a key technical step and we believe is of independent interest.

► **Theorem 10 (♠).** *Given a STRICT-UETR instance $\Phi := \forall X \in \mathbb{R}^n . \exists Y \in \mathbb{R}^m : \varphi_{<}(X, Y)$, with $\varphi_{<}(X, Y) \in \text{QFF}_{<}$, we can construct in polynomial time an equivalent UETR instance*

$$\Psi := \forall X \in [-1, 1]^n . \exists Y \in [-1, 1]^\ell : \psi(X, Y),$$

where $\psi \in \text{QFF}$. Further, $\perp(\Psi)$ is either empty or contains an n -dimensional open ball.

5 $\forall\exists_{<}\mathbb{R}$ -Hardness

► **Theorem 11.** HAUSDORFF and directed HAUSDORFF are $\forall\exists_{<}\mathbb{R}$ -hard.

Proof. Let $\Phi := \forall X \in \mathbb{R}^n . \exists Y \in \mathbb{R}^m : \varphi_{<}(X, Y)$ be an instance of STRICT-UETR. We give a polynomial-time many-one reduction to an equivalent HAUSDORFF instance. The proof is split into three parts: First we transform Φ into an equivalent UETR instance Ψ' whose counterexamples contain an open ball (if there are any). Then we use Ψ' to define the semi-algebraic sets A and B as well as an integer t , such that (A, B, t) is a HAUSDORFF instance. Lastly we prove that Φ and (A, B, t) are indeed equivalent.

Transforming Φ into Ψ' . We apply Theorem 10 to Φ and obtain an equivalent sentence

$$\Psi := \forall X \in [-1, 1]^n . \exists Y \in [-1, 1]^\ell : \psi(X, Y)$$

in polynomial time, where $\psi(X, Y) \in \text{QFF}$. Additionally, we get that $\perp(\Psi) = \emptyset$ if Ψ is true and that it contains an n -dimensional open ball $B_n(x, r)$ centered at some $x \in \perp(\Psi) \subseteq [-1, 1]^n$ of radius $r > 0$ otherwise. We remark that Ψ is an instance of UETR and not necessarily of STRICT-UETR. Using the tools from Section 3, we shall prove next, that we can give a lower bound on the radius r of the open ball of counterexamples centered at x . For this, assume that Ψ is false, so $\perp(\Psi) \neq \emptyset$ and therefore

$$\neg\Psi = \exists X \in [-1, 1]^n . \forall Y \in [-1, 1]^\ell : \neg\psi(X, Y)$$

is true. Utilizing our knowledge about the open ball of counterexamples around x , we can strengthen this to

$$\exists r > 0 . \exists X \in [-1, 1]^n . \forall \tilde{X} \in [-1, 1]^n, Y \in [-1, 1]^\ell : \|X - \tilde{X}\|^2 < r^2 \implies \neg\psi(\tilde{X}, Y),$$

which is still equivalent to $\neg\Psi$. Let L denote the length of the quantifier-free part of this formula. We see that L is clearly polynomial in $|\Psi|$, which by Theorem 10 is polynomial in $|\Phi|$. The above sentence has the form required to apply Lemma 8, and we get that there is an r satisfying above sentence with

$$r \geq 2^{-L\beta^4 n(n+\ell)} \tag{1}$$

for some constant $\beta \in \mathbb{R}$. Let N be the smallest integer, such that

$$r \cdot 2^{2^N} > \ell. \tag{2}$$

By Equation (1), it holds that $N \in O(n(n + \ell) \log(L))$. Using Lemma 9 on Ψ and N , we can again in polynomial time scale up the range of the universally quantified variables and get

$$\Psi' := \forall X \in [-2^{2^N}, 2^{2^N}]^n . \exists Y \in [-1, 1]^\ell, U \in [-1, 1]^{N+1} : \psi'(X, Y, U),$$

where $\psi'(X, Y, U) \in \text{QFF}$ and we have $\perp(\Psi')$ equal to $\perp(\Psi)$ scaled up by 2^{2^N} in all dimensions. Further, from (the proof of) Lemma 9 it follows and for all $(x, y, u) \in \mathbb{R}^{n+\ell+N+1}$ with $\psi'(x, y, u)$ we have $u_i = 2^{-2^i}$. In particular, the radius of the open ball of counterexamples around $2^{2^N} \cdot x \in \perp(\Psi')$ is now $r' := r \cdot 2^{2^N} > \ell$ by the choice of N .

Defining HAUSDORFF instance (A, B, t) . We first define three sets A', B' and C' as follows:

$$\begin{aligned} A' &:= \{(x, y, u) \in [-2^{2^N}, 2^{2^N}]^n \times [-1, 1]^\ell \times [-1, 1]^{N+1} \mid \psi'(x, y, u)\} \\ B' &:= [-2^{2^N}, 2^{2^N}]^n \times \{0\}^\ell \times \{2^{-2^0}\} \times \dots \times \{2^{-2^N}\} \\ C' &:= \{2^{2^{N+1}}\}^{n+\ell} \times \{2^{-2^0}\} \times \dots \times \{2^{-2^N}\} \end{aligned}$$

Note that $A', B', C' \subseteq \mathbb{R}^{n+\ell+N+1}$ and all three sets can be described by quantifier-free formulas of polynomial length. We further define

$$\begin{aligned} A &:= A' \cup C', \\ B &:= B' \cup C' \quad \text{and} \\ t &:= \ell. \end{aligned}$$

The reason to include C' into both A and B is to guarantee that both semi-algebraic sets are non-empty. Otherwise, if $\perp(\Psi') = [-2^{2^N}, 2^{2^N}]^n$, the set A is the empty set and the Hausdorff distance between A and B would not be well-defined. The triple (A, B, t) is the desired HAUSDORFF instance.

Equivalence of Φ and (A, B, t) . We first note that we can ignore C' in our argumentation about $d_H(A, B)$: In fact, assuming that both A' and B' are non-empty, we have $d_H(A, B) = d_H(A', B')$. To prove this, observe first that adding the same set of points to A' and B' can only decrease their Hausdorff distance. Second, C' was chosen to have $d_H(A', C') \geq d_H(A', B')$, so for no $a \in A$, the distance to the closest $b \in B$ has decreased (and vice versa).

To see that Φ and (A, B, t) are equivalent, assume first that Φ is true. Let $u \in [-1, 1]^{N+1}$ such that $u_i = 2^{-2^i}$. As seen above, this is necessary in every satisfying assignment of the variable vector U in Ψ' . Then for every $x \in [-2^{2^N}, 2^{2^N}]^n$ there is at least one $y \in [-1, 1]^\ell$ such that $a = (x, y, u) \in A$. At the same time, $b = (x, \{0\}^\ell, u) \in B$. We get

$$\|a - b\| = \|(x, y, u) - (x, \{0\}^\ell, u)\| = \|y - \vec{0}\| \leq \sqrt{\sum_{i=1}^\ell 1} = \sqrt{\ell} \leq \ell = t.$$

As x was chosen arbitrarily, we get an upper bound for the directed Hausdorff distance $\vec{d}_H(A, B) \leq \ell$. On the other hand, for every $b = (x, \{0\}^\ell, u) \in B$ there is an $y \in [-1, 1]^\ell$ such that $a = (x, y, u) \in A$, as we assume that Φ is true. As above, we get $\vec{d}_H(B, A) \leq \ell$ and thus

$$d_H(A, B) \leq \ell = t. \tag{3}$$

Now assume that Φ is false. By construction Ψ' is also false and contains a counterexample $x \in \perp(\Psi')$ such that $B_n(x, r') \subseteq \perp(\Psi')$. Consider $b = (x, \{0\}^\ell, u) \in B$. Since Ψ' is false, for no \tilde{x} with $\|x - \tilde{x}\| < r'$ and no $y \in [-1, 1]^\ell$ there is a point $a = (\tilde{x}, y, u) \in A$. We conclude

$$d_H(A, B) \geq \vec{d}_H(B, A) \geq r' > \ell = t. \tag{4}$$

Equations (3) and (4) prove that $d_H(A, B) \leq t$ (and $\vec{d}_H(B, A) \leq t$) if and only if Φ is true. ◀

In the proof of Theorem 1, we could choose $N' := N + 1$ instead of N in Equation (2). Then in the case that Φ is false, the Hausdorff distance is at least

$$r' > 2^{2^{N+1}} r > 2^{2^{N+1}-2^N} \ell = 2^{2^N} \ell = 2^{2^N} t.$$

Note that the dimension d of the resulting sets A, B equals $d = n + \ell + N' + 1 = \Theta(N)$. Thus, we created a gap of size $2^{2^{\Theta(d)}}$. This implies the following inapproximability result.

► **Corollary 2.** *Let A and B be two semi-algebraic sets in \mathbb{R}^d and $f(d) = 2^{2^{o(d)}}$. Then there is no polynomial-time $f(d)$ -approximation algorithm to compute $d_H(A, B)$, unless $P = \forall \exists < \mathbb{R}$.*

6 Open problems

We showed that the HAUSDORFF problem is $\forall \exists < \mathbb{R}$ complete. One important open question is whether the two complexity classes $\forall \exists \mathbb{R}$ and $\forall \exists < \mathbb{R}$ are actually the same. An answer to this question is interesting in its own right. Furthermore, it is interesting to see if our hardness result can be extended to simpler settings.

References

- 1 Zachary Abel, Erik Demaine, Martin Demaine, Sarah Eisenstat, Jayson Lynch, and Tao Schardl. Who Needs Crossings? Hardness of Plane Graph Rigidity. In Sándor Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry (SoCG 2016)*, volume 51 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:15, 2016. doi:10.4230/LIPIcs.SocG.2016.3.
- 2 Mikkel Abrahamsen. Covering Polygons is Even Harder. arXiv preprint, 2021. arXiv:2106.02335.
- 3 Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. The Art Gallery Problem is $\exists \mathbb{R}$ -complete. In *STOC 2018: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 65–73, 2018. doi:10.1145/3188745.3188868.
- 4 Mikkel Abrahamsen, Linda Kleist, and Tillmann Miltzow. Geometric Embeddability of Complexes is $\exists \mathbb{R}$ -complete. arXiv preprint, 2021. arXiv:2108.02585.
- 5 Mikkel Abrahamsen, Linda Kleist, and Tillmann Miltzow. Training Neural Networks is ER-complete. In Marc A. Ranzato, Alina Beygelzimer, K. Nguyen, Percy Liang, Jennifer W. Vaughan, and Yann Dauphin, editors, *Advances in Neural Information Processing Systems (NeurIPS 2021)*, volume 34, 2021.
- 6 Mikkel Abrahamsen, Tillmann Miltzow, and Nadja Seiferth. Framework for ER-Completeness of Two-Dimensional Packing Problems. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1014–1021, 2020. doi:10.1109/FOCS46700.2020.00098.
- 7 Helmut Alt, Bernd Behrends, and Johannes Blömer. Approximate Matching of Polygonal Shapes. *Annals of Mathematics and Artificial Intelligence*, 13(3):251–265, 1995. doi:10.1007/BF01530830.
- 8 Helmut Alt, Peter Braß, Michael Godau, Christian Knauer, and Carola Wenk. Computing the Hausdorff Distance of Geometric Patterns and Shapes. In Boris Aronov, Saugata Basu, János Pach, and Micha Sharir, editors, *Discrete and Computational Geometry: The Goodman-Pollack Festschrift*, volume 25 of *Algorithms and Combinatorics*, pages 65–76. Springer, 2003. doi:10.1007/978-3-642-55566-4_4.
- 9 Helmut Alt and Leonidas J. Guibas. Discrete Geometric Shapes: Matching, Interpolation, and Approximation. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 121–153. Elsevier, 2000. doi:B978-044482537-7/50004-8.
- 10 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. doi:10.1017/CB09780511804090.

- 11 Mikhail J. Atallah. A Linear Time Algorithm for the Hausdorff Distance Between Convex Polygons. *Information Processing Letters*, 17(4):207–209, 1983. doi:10.1016/0020-0190(83)90042-X.
- 12 Sauguta Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer, 2006. doi:10.1007/3-540-33099-2.
- 13 Sauguta Basu and Marie-Françoise Roy. Bounding the radii of balls meeting every connected component of semi-algebraic sets. *Journal of Symbolic Computation*, 45(12):1270–1279, 2010. doi:10.1016/j.jsc.2010.06.009.
- 14 Maria Belk. Realizability of Graphs in Three Dimensions. *Discrete & Computational Geometry*, 37(2):139–162, 2007. doi:10.1007/s00454-006-1285-4.
- 15 Vittorio Bilò and Marios Mavronicolas. A Catalog of EXISTS-R-Complete Decision Problems About Nash Equilibria in Multi-Player Games. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 17:1–17:13, 2016. doi:10.4230/LIPIcs.STACS.2016.17.
- 16 Lenore Blum, Mike Shub, and Steve Smale. On a Theory of Computation and Complexity over the Real Numbers: NP-Completeness, Recursive Functions and Universal Machines. *Bulletin of the American Mathematical Society*, 21:1–46, 1989. doi:10.1090/S0273-0979-1989-15750-9.
- 17 Glen E. Bredon. *Topology and Geometry*, volume 139 of *Graduate Texts in Mathematics*. Springer Science & Business Media, 1st edition, 2013. doi:10.1007/978-1-4757-6848-0.
- 18 Karl Bringmann and André Nusser. Translating Hausdorff Is Hard: Fine-Grained Lower Bounds for Hausdorff Distance Under Translation. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:17, 2021. doi:10.4230/LIPIcs.SocG.2021.18.
- 19 Peter Bürgisser and Felipe Cucker. Exotic Quantifiers, Complexity Classes, and Complete Problems. *Foundations of Computational Mathematics*, 9(2):135–170, 2009. doi:10.1007/s10208-007-9006-9.
- 20 Jean Cardinal. Computational Geometry Column 62. *ACM SIGACT News*, 46(4):69–78, 2015. doi:10.1145/2852040.2852053.
- 21 Jean Cardinal and Udo Hoffmann. Recognition and Complexity of Point Visibility Graphs. *Discrete & Computational Geometry*, 57(1):164–178, 2017. doi:10.1007/s00454-016-9831-1.
- 22 Wiki Community. Homotopy. accessed 2021 November. URL: <https://en.wikipedia.org/wiki/Homotopy>.
- 23 David Cox, John Little, and Donal O’Shea. *Using Algebraic Geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 2006. doi:10.1007/b138611.
- 24 James H. Davenport and Joos Heintz. Real Quantifier Elimination is Doubly Exponential. *Journal of Symbolic Computation*, 5(1–2):29–35, 1988. doi:10.1016/S0747-7171(88)80004-X.
- 25 Julian D’Costa, Engel Lefauchaux, Eike Neumann, Joël Ouaknine, and James Worrel. On the Complexity of the Escape Problem for Linear Dynamical Systems over Compact Semialgebraic Sets. In Filippo Bonchi and Simon J. Puglisi, editors, *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 202 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:21, 2021. doi:10.4230/LIPIcs.MFCS.2021.33.
- 26 Michael G. Dobbins, Andreas Holmsen, and Tillmann Miltzow. A Universality Theorem for Nested Polytopes. arXiv preprint, 2019. arXiv:1908.02213.
- 27 Michael G. Dobbins, Linda Kleist, Tillmann Miltzow, and Paweł Rzażewski. $\forall\exists\mathbb{R}$ -Completeness and Area-Universality. In Andreas Brandstädt, Ekkehard Köhler, and Klaus Meer, editors, *Graph-Theoretic Concepts in Computer Science (WG)*, volume 11159 of *Lecture Notes in Computer Science*, pages 164–175. Springer, 2018. doi:10.1007/978-3-030-00256-5_14.
- 28 Michael G. Dobbins, Linda Kleist, Tillmann Miltzow, and Paweł Rzażewski. Completeness for the Complexity Class $\forall\exists\mathbb{R}$ and Area-Universality. arXiv preprint, 2021. arXiv:1712.05142v3.


- 29 Jugal Garg, Ruta Mehta, Vijay V. Vazirani, and Sadra Yazdanbod. $\exists\mathbb{R}$ -Completeness for Decision Versions of Multi-Player (Symmetric) Nash Equilibria. *ACM Transactions on Economics and Computation*, 6(1):1:1–1:23, 2018. doi:10.1145/3175494.
- 30 Michael Godau. *On the complexity of measuring the similarity between geometric objects in higher dimensions*. PhD thesis, Freie Universität Berlin, 1999. doi:10.17169/refubium-7780.
- 31 Dmitrii Y. Grigor'ev and Nicolai N. Vorobjov. Solving Systems of Polynomial Inequalities in Subexponential Time. *Journal of Symbolic Computation*, 5(1-2):37–64, 1988. doi:10.1016/S0747-7171(88)80005-1.
- 32 Felix Hausdorff. *Grundzüge der Mengenlehre*. Von Veit & Company, 1914.
- 33 Anna Lubiw, Tillmann Miltzow, and Debajyoti Mondal. The Complexity of Drawing a Graph in a Polygonal Region. In Therese Biedl and Andreas Kerren, editors, *GD 2018: Graph Drawing and Network Visualization*, volume 11282 of *Lecture Notes in Computer Science*, pages 387–401, 2018. doi:10.1007/978-3-030-04414-5_28.
- 34 Jiří Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. arXiv preprint, 2014. arXiv:1406.2636.
- 35 Colin McDiarmid and Tobias Müller. Integer realizations of disk and segment graphs. *Journal of Combinatorial Theory, Series B*, 103(1):114–143, 2013. doi:10.1016/j.jctb.2012.09.004.
- 36 Tillmann Miltzow and Reinier F. Schmiermann. On Classifying Continuous Constraint Satisfaction Problems. arXiv preprint, 2022. arXiv:2106.02397.
- 37 Deng Min, Li Zhilin, and Chen Xiaoyong. Extended Hausdorff distance for spatial objects in GIS. *International Journal of Geographical Information Science*, 21(4):459–475, 2007. doi:10.1080/13658810601073315.
- 38 Nikolai E. Mnëv. The Universality Theorems on the Classification Problem of Configuration Varieties and Convex Polytopes Varieties. In Oleg Y. Viro and Anatoly M Vershik, editors, *Topology and Geometry — Rohlin Seminar*, volume 1346 of *Lecture Notes in Mathematics*, pages 527–543. Springer, 1988. doi:10.1007/BFb0082792.
- 39 Tim Ophelders, Ignaz Rutter, Bettina Speckmann, and Kevin Verbeek. Polygon-Universal Graphs. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 55:1–55:15, 2021. doi:10.4230/LIPIcs.SoCG.2021.55.
- 40 James Renegar. On the Computational Complexity and Geometry of the First-order Theory of the Reals. Part I: Introduction. Preliminaries. The Geometry of Semi-algebraic Sets. The Decision Problem for the Existential Theory of the Reals. *Journal of Symbolic Computation*, 13(3):255–299, 1992. doi:10.1016/S0747-7171(10)80003-3.
- 41 James Renegar. On the Computational Complexity and Geometry of the First-Order Theory of the Reals. Part II: The General Decision Problem. Preliminaries for Quantifier Elimination. *Journal of Symbolic Computation*, 13(3):301–327, 1992. doi:10.1016/S0747-7171(10)80004-5.
- 42 James Renegar. On the Computational Complexity and Geometry of the First-Order Theory of the Reals. Part III: Quantifier Elimination. *Journal of Symbolic Computation*, 13(3):329–352, 1992. doi:10.1016/S0747-7171(10)80005-7.
- 43 Jürgen Richter-Gebert and Günter M. Ziegler. Realization Spaces of 4-Polytopes are Universal. *Bulletin of the American Mathematical Society*, 32(4):403–412, 1995. doi:10.1090/S0273-0979-1995-00604-X.
- 44 Tim Roughgarden. Beyond Worst-Case Analysis. *Communications of the ACM*, 62(3):88–96, 2019. doi:10.1145/3232535.
- 45 William Rucklidge. *Efficient Visual Recognition Using the Hausdorff Distance*, volume 1173 of *Lecture Notes in Computer Science*. Springer, 1996. doi:10.1007/BFb0015091.
- 46 Marcus Schaefer. Complexity of Some Geometric and Topological Problems. In David Eppstein and Emden R. Gansner, editors, *GD 2009: Graph Drawing*, volume 5849 of *Lecture Notes in Computer Science*, pages 334–344, 2010. doi:10.1007/978-3-642-11805-0_32.
- 47 Marcus Schaefer. *Realizability of Graphs and Linkages*, pages 461–482. Thirty Essays on Geometric Graph Theory. Springer, 2013. doi:10.1007/978-1-4614-0110-0_24.

- 48 Marcus Schaefer and Daniel Štefankovič. Fixed Points, Nash Equilibria, and the Existential Theory of the Reals. *Theory of Computing Systems*, 60:172–193, 2017. doi:10.1007/s00224-015-9662-0.
- 49 Yaroslav Shitov. A Universality Theorem for Nonnegative Matrix Factorizations. arXiv preprint, 2016. arXiv:1606.09068.
- 50 Peter W. Shor. Stretchability of Pseudolines is NP-Hard. In Peter Gritzmann and Bernd Sturmfels, editors, *Applied Geometry And Discrete Mathematics*, volume 4 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 531–554, 1991. doi:10.1090/dimacs/004/41.
- 51 Daniel Spielman and Shang-Hua Teng. Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time. *Journal of the ACM*, 51(3):385–463, 2004. doi:10.1145/990308.990310.
- 52 Floor Verhoeven, Amir Vaxman, Tim Hoffmann, and Olga Sorkine-Hornung. Dev2PQ: Planar Quadrilateral Strip Remeshing of Developable Surfaces. *ACM Transactions on Graphics*, 41(3):29:1–29:18, 2022. doi:10.1145/3510002.
- 53 Nicolai N. Vorob'ev. Estimates of Real Roots of a System of Algebraic Equations. *Journal of Soviet Mathematics*, 34:1754–1762, 1986. doi:10.1007/BF01095637.

Dynamic Connectivity in Disk Graphs

Haim Kaplan ✉

School of Computer Science,
Tel Aviv University, Israel

Katharina Klost ✉ 

Institut für Informatik,
Freie Universität Berlin, Germany

Wolfgang Mulzer ✉ 

Institut für Informatik,
Freie Universität Berlin, Germany

Paul Seiferth ✉

Institut für Informatik,
Freie Universität Berlin, Germany

Alexander Kauer ✉

Institut für Informatik,
Freie Universität Berlin, Germany

Kristin Knorr ✉ 

Institut für Informatik,
Freie Universität Berlin, Germany

Liam Roditty ✉

Department of Computer Science,
Bar Ilan University, Ramat Gan, Israel

Abstract

Let $S \subseteq \mathbb{R}^2$ be a set of n planar *sites*, such that each $s \in S$ has an *associated radius* $r_s > 0$. Let $\mathcal{D}(S)$ be the *disk intersection graph* for S . It has vertex set S and an edge between two distinct sites $s, t \in S$ if and only if the disks with centers s, t and radii r_s, r_t intersect. Our goal is to design data structures that maintain the connectivity structure of $\mathcal{D}(S)$ as sites are inserted and/or deleted.

First, we consider *unit disk graphs*, i.e., $r_s = 1$, for all $s \in S$. We describe a data structure that has $O(\log^2 n)$ amortized update and $O(\log n / \log \log n)$ amortized query time. Second, we look at disk graphs *with bounded radius ratio* Ψ , i.e., for all $s \in S$, we have $1 \leq r_s \leq \Psi$, for a $\Psi \geq 1$ known in advance. In the fully dynamic case, we achieve amortized update time $O(\Psi \lambda_6(\log n) \log^7 n)$ and query time $O(\log n / \log \log n)$, where $\lambda_s(n)$ is the maximum length of a Davenport-Schinzel sequence of order s on n symbols. In the incremental case, where only insertions are allowed, we get logarithmic dependency on Ψ , with $O(\alpha(n))$ query time and $O(\log \Psi \lambda_6(\log n) \log^7 n)$ update time. For the decremental setting, where only deletions are allowed, we first develop an efficient *disk revealing* structure: given two sets R and B of disks, we can delete disks from R , and upon each deletion, we receive a list of all disks in B that no longer intersect the union of R . Using this, we get decremental data structures with amortized query time $O(\log n / \log \log n)$ that support m deletions in $O((n \log^5 n + m \log^7 n) \lambda_6(\log n) + n \log \Psi \log^4 n)$ overall time for bounded radius ratio Ψ and $O((n \log^6 n + m \log^8 n) \lambda_6(\log n))$ for arbitrary radii.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Data structures design and analysis; Mathematics of computing → Paths and connectivity problems

Keywords and phrases Disk Graphs, Connectivity, Lower Envelopes

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.49

Related Version *Full Version*: <https://arxiv.org/abs/2106.14935>

Funding Supported in part by grant 1367/2016 from the German-Israeli Science Foundation (GIF). *Haim Kaplan*: Partially supported by ISF grant 1595/19 and by the Blavatnik research foundation. *Alexander Kauer*: Supported in part by grant 1367/2016 from the German-Israeli Science Foundation (GIF), by the German Research Foundation within the collaborative DACH project *Arrangements and Drawings* as DFG Project MU 3501/3-1, and by ERC StG 757609.

Kristin Knorr: Supported by the German Science Foundation within the research training group “Facets of Complexity” (GRK 2434).

Wolfgang Mulzer: Supported in part by ERC StG 757609.



© Haim Kaplan, Alexander Kauer, Katharina Klost, Kristin Knorr, Wolfgang Mulzer, Liam Roditty, and Paul Seiferth;
licensed under Creative Commons License CC-BY 4.0

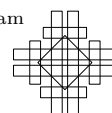
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 49; pp. 49:1–49:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Suppose we are given a simple, undirected graph G , and we would like to preprocess it so that we can determine efficiently if two vertices of G lie in the same connected component. If G is fixed, we can simply perform a graph search in G (e.g., BFS or DFS) to label the vertices of each connected component with a unique identifier, allowing us to answer all queries in $O(1)$ time with linear preprocessing time and space. When G changes over time, the problem becomes much harder. If the vertex set is fixed and edges can only be inserted, the problem reduces to disjoint set union. Then, there is a folklore optimal data structure. It achieves $O(1)$ time for updates and $O(\alpha(n))$ amortized time for queries, where $\alpha(n)$ is the inverse Ackermann function [5]. If the vertex set is fixed, but edges can be inserted and deleted, there is a data structure due to Holm et al. [8], with $O(\log n / \log \log n)$ amortized query time and $O(\log^2 n)$ amortized update time. For planar graphs, Eppstein et al. [6] give a structure with $O(\log n)$ amortized time for queries *and* updates.

In this paper, we add a geometric twist and study the dynamic connectivity problem on different variants of *disk intersection graphs*. Let $S \subset \mathbb{R}^2$ be a set of planar *point sites*, where each site $s \in S$ has an associated radius $r_s > 0$. The *disk intersection graph* (*disk graph*, for short) $\mathcal{D}(S)$ is the undirected graph with vertex set S that has an undirected edge between any two distinct sites s and t if and only if the Euclidean distance between s and t is at most $r_s + r_t$. Note that even though $\mathcal{D}(S)$ is fully described by the n sites and their associated radii, it might have $\Theta(n^2)$ edges. Thus, our goal is to find algorithms whose running time depends only on the number of sites and not on the number of edges. We consider three variants of disk graphs, characterized by the possible values for the radii. In the first variant, *unit disk graphs*, all radii are 1. In the second variant, *bounded radius ratio*, all radii must come from the interval $[1, \Psi]$, where Ψ is a parameter known in advance that may depend on the number of sites n . In the third variant, *general disk graphs*, the radii can be arbitrary.

We assume that S is dynamic, i.e., sites can be inserted and deleted over time. At each update, the edges incident to the modified site appear or disappear in $\mathcal{D}(S)$. An update can change up to $n - 1$ edges in $\mathcal{D}(S)$, so simply storing $\mathcal{D}(S)$ in the data structure by Holm et al. could lead to potentially superlinear update times and might even be slower than recomputing the connectivity information from scratch.

For dynamic connectivity in general disk graphs, Chan et al. [4] give a data structure with amortized $O(n^{1/7+\epsilon})$ query time and $O(n^{20/21+\epsilon})$ update time. As far as we know, this is still the currently best fully dynamic connectivity structure for general disk graphs. However, Chan et al. present their data structure as a special case of a more general setting, so there is hope that the specific geometry of disk graphs may allow for better running times.

Indeed, several results show that for certain disk graphs, we can achieve polylogarithmic update and query times. For unit disk graphs, Chan et al. [4] observe that there is a data structure with $O(\log^6 n)$ update time and $O(\log n / \log \log n)$ query time.¹ For bounded radius ratio, Kaplan et al. [9] show that there is a data structure with expected amortized update time $O(\Psi^2 \lambda_6(\log n) \log^7 n)$ and query time $O(\log n / \log \log n)$.² Both results use the notion of a *proxy graph*, a sparse graph that models the connectivity of the original disk graph and that can be updated efficiently with suitable dynamic geometric data structures. The proxy graph can then be stored in the data structure by Holm et al., so the query procedure coincides with the one by Holm et al. The update operations involve a combination of updating the proxy graph with the help of the geometric data structures and of modifying the edges in the structure of Holm et al.

¹ Actually, Chan et al. [4] claim an update time of $O(\log^{10} n)$. Recent results [3] improve the bound.

² The original paper claims an update time of $O(\Psi^2 \lambda_6(\log n) \log^9 n)$, but recent improvements in the underlying data structure [10] lead to the better bound.

Our results. For unit disk graphs, we significantly improve over Chan et al. [4]: with a direct approach that uses a grid-based proxy graph and dynamic lower envelopes, we obtain $O(\log^2 n)$ amortized update and $O(\log n / \log \log n)$ amortized query time (Theorem 3.2).

For bounded radius ratio, we give a data structure that improves the update time. Specifically, we achieve expected amortized update time $O(\Psi \lambda_6(\log n) \log^7 n)$ and amortized query time $O(\log n / \log \log n)$, where $\lambda_s(n)$ is the maximum length of a Davenport-Schinzel sequence of order s on n symbols [11]. Compared to the previous data structure of Kaplan et al., this improves the factor in the update time from Ψ^2 to Ψ .

We also provide partial results that push the dependency on Ψ from linear to logarithmic. For this, we consider the *semi-dynamic* setting, in which only insertions (*incremental*) or only deletions (*decremental*) are allowed. In the incremental setting, we use a dynamic additively weighted Voronoi diagram to obtain a data structure with $O(\alpha(n))$ amortized query time and $O(\log \Psi \lambda_6(\log n) \log^7 n)$ expected amortized update time. Due to space reasons, this result is deferred to the full version. In the decremental setting, a main challenge is to identify those edges in $\mathcal{D}(S)$ that were incident to a freshly removed site and that change the connectivity in $\mathcal{D}(S)$. To address this, we first develop a data structure for a related dynamic geometric problem which might be of independent interest: suppose we have two sets R and B of disks in the plane, such that the disks in B can only be deleted, while the disks in R can be both inserted and deleted. We would like to maintain R and B in a data structure such that whenever we delete a disk b from B , we receive a list of all the disks in the current set R that intersect the disk b but no other disk from the remaining set $B \setminus \{b\}$. We say that these are the disks in R that are *revealed* by the deletion of b . We call this data structure a *disk revealing structure* (RDS). Due to space reasons, the details of the RDS are relegated to the full version. Its properties are summarized in the following theorem:

► **Theorem 1.1.** *Let R and B be disjoint sets of disks in \mathbb{R}^2 with $|R| + |B| = n$. We can preprocess $R \cup B$ into a structure that supports deletions from $R \cup B$, while detecting all newly revealed disks of R after each deletion from B . Preprocessing needs $O(|B| \log^5 n \lambda_6(\log n) + |R| \log^3 n)$ expected time and $O(n \log n)$ expected space. Deleting k disks from B and any number of disks from R needs $O(|R| \log^4 n + k \log^7 n \lambda_6(\log n))$ expected time, where $\lambda_s(n)$ is the maximum length of a Davenport-Schinzel sequence of order s .*

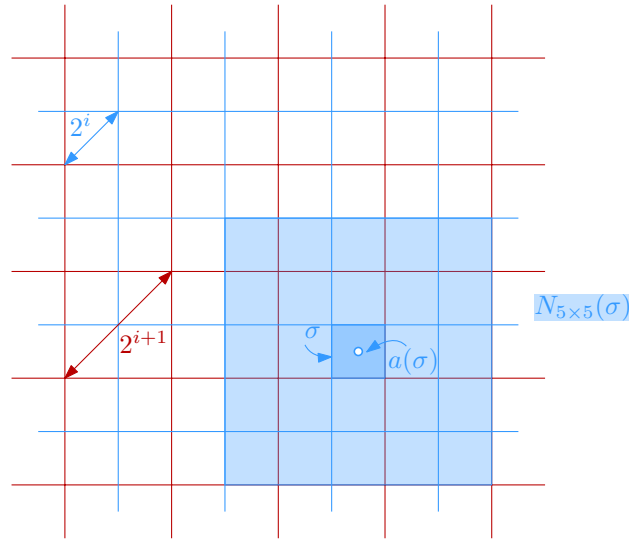
The RDS plays a crucial part in developing decremental connectivity structures for disk graphs of bounded radius ratio and for general disk graphs. For both cases, we obtain data structures with $O(\log n / \log \log n)$ amortized query time. The total expected time for processing k deletions is $O((n \log^5 n + k \log^7 n) \lambda_6(\log n) + n \log \Psi \log^4 n)$ for bounded radius ratio (Theorem 5.6) and $O((n \log^6 n + k \log^8 n) \lambda_6(\log n))$ for the general case (full version).

2 Preliminaries

Data structure for edge updates. We rely on the following existing data structure that supports connectivity queries and edge updates on general graphs.

► **Theorem 2.1** (Holm et al. [8, Theorem 3]). *Let G be a graph with n vertices and initially no edges. There is a deterministic fully dynamic data structure so that edge updates in G take amortized time $O(\log^2 n)$ and connectivity queries take worst-case time $O(\log n / \log \log n)$.*

Theorem 2.1 assumes that n is fixed, but we can easily support vertex insertions and deletions within the same amortized time bounds, with standard rebuilding. Thorup gave a variant of Theorem 2.1 that uses $O(m)$ space, where m is the current number of edges [13].



■ **Figure 1** Two levels of the hierarchical grid.

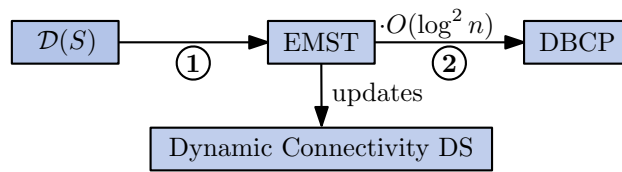
The hierarchical grid and quadtrees. Let \mathcal{G}_i be a grid with cell diameter 2^i and a corner at the origin. The *hierarchical grid* \mathcal{G} is defined as $\mathcal{G} = \bigcup_{i=0}^{\infty} \mathcal{G}_i$. For any cell $\sigma \in \mathcal{G}$, we denote by $|\sigma|$ its diameter and by $a(\sigma)$ its center. We say that grid \mathcal{G}_i has *level* i . We assume that we can find the coordinates of the cell of \mathcal{G} containing a site on a given level in $O(1)$ time. Furthermore, for a cell $\sigma \in \mathcal{G}_i$ and odd k , we call the $k \times k$ subgrid of \mathcal{G}_i centered at σ the $(k \times k)$ -*neighborhood* of σ , and denote it by $N_{k \times k}(\sigma)$; see Figure 1. Let \mathcal{C} be a set of cells in \mathcal{G} . The *quadtree* \mathcal{T} for \mathcal{C} is a rooted 4-nary tree whose nodes are cells from \mathcal{G} . The root of \mathcal{C} is the smallest cell ρ in \mathcal{G} that contains all of \mathcal{C} . If a cell σ with $|\sigma| = 2^i$, for $i \geq 1$, properly contains at least one cell of \mathcal{C} , then the four children of σ are the cells τ with $|\tau| = 2^{i-1}$ and $\tau \subseteq \sigma$. If a cell σ does not properly contain a cell of \mathcal{C} , it does not have any children. Typically, we do not distinguish between a cell σ and its associated vertex. A quadtree \mathcal{T} on a given set of n cells can be constructed in $O(n \log(|\rho|))$ time, where ρ is the root of \mathcal{T} .

Maximal bichromatic matchings. We need a data structure that dynamically maintains a *maximal bichromatic matching* (MBM) between two sets of disks: let $R \subseteq S$ and $B \subseteq S$ be two disjoint non-empty sets of sites, and $(R \times B) \cap \mathcal{D}(S)$ the bipartite graph on R and B with all edges of $\mathcal{D}(S)$ with one vertex in R and one vertex in B . An MBM between R and B is a maximal set of vertex-disjoint edges in $(R \times B) \cap \mathcal{D}(S)$. We show how to maintain an MBM as sites are inserted or deleted in R and in B , in two ways. The first way uses a general structure by Kaplan et al. [9] and applies in all settings, see the full version for details.

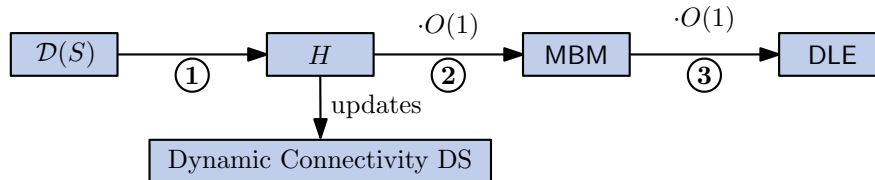
► **Lemma 2.2.** *Let $R, B \subseteq S$ be two disjoint sets with a total of at most n sites. Then, there exists a dynamic data structure that maintains an MBM for R and B with $O(\lambda_6(\log n) \log^7 n)$ amortized expected update time, using $O(n \log n)$ expected space.*

The second way applies only to unit disks that are separated by a vertical or horizontal lines. It relies on dynamic lower envelopes for pseudolines [1], see the full version for details.

► **Lemma 2.3.** *Suppose that $r_s = 1$, for all sites $s \in S$. Let $R, B \subseteq S$ be two disjoint sets with a total of at most n sites, such that there is a known vertical or horizontal line that separates R and B . Then, there exists a dynamic data structure that maintains an MBM for R and B with $O(\log^2 n)$ worst-case update time, using $O(n)$ space.*



■ **Figure 2** A solution with $O(\log^6 n)$ update time.



■ **Figure 3** The structure for our data structure.

3 Fully dynamic unit disk graphs

We first consider the case of unit disk graphs. As mentioned in the introduction, this problem was already addressed by Chan et al. [4]. They explained how to combine several known results into a data structure for connectivity queries in fully dynamic unit disk graphs with update time $O(\log^6 n)$ and query time $O(\log n / \log \log n)$.

A visual representation of their approach can be found in Figure 2. In the core, they use a subtree of the Euclidean minimum spanning tree (EMST) of S as a *proxy graph* that accurately represents the connectivity of $\mathcal{D}(S)$. They store this proxy graph in a Holm et al. data structure. In order to update the EMST efficiently, they also maintain several instances of a *dynamic bichromatic closest pair* problem (DBCP). The combination of the running times for the separated data structures then yields the overall running time claimed above. To improve over this result, we replace the EMST by a simpler graph that still captures the connectivity of $\mathcal{D}(S)$. We also replace the DBCP structure by a suitable maximal bichromatic matching (MBM) structure that is based on dynamic lower envelopes (Lemma 2.3). These two changes significantly improve the amortized update time to $O(\log^2 n)$, without affecting the query time. The overall structure behind our method is shown in Figure 3.

We define a *proxy graph* H that represents the connectivity of $\mathcal{D}(S)$. The vertices of H are cells of the grid \mathcal{G}_1 of diameter 2 (cf. Section 2). More precisely, we say that two cells σ, τ in \mathcal{G}_1 are *neighboring* if $\sigma \in N_{5 \times 5}(\tau)$. For $S \subset \mathbb{R}^2$, we define the graph H whose vertices are the *non-empty* cells $\sigma \in \mathcal{G}_1$, i.e., the cells with $\sigma \cap S \neq \emptyset$. We say that a site $s \in S$ is *assigned* to the cell $\sigma \in \mathcal{G}_1$ that contains it, and we let $S(\sigma)$ denote the sites that are assigned to σ . Two cells σ, τ are connected by an edge in H if and only if there is an edge $st \in \mathcal{D}(S)$ with $s \in S(\sigma)$ and $t \in S(\tau)$. Then, H is sparse and represents the connectivity in $\mathcal{D}(S)$, as stated in the following lemma. Its simple proof can be found in the full version.

► **Lemma 3.1.** *The proxy graph H has at most n vertices, each with degree $O(1)$. Two sites $s, t \in S$ are connected in $\mathcal{D}(S)$ if and only if their assigned cells σ and τ are connected in H .*

We build a data structure \mathcal{H} as in Theorem 2.1 for H . To query the connectivity between two sites s and t , we first identify the cells σ and τ in \mathcal{G}_1 to which s and t are assigned. This requires $O(1)$ time, because when inserting a site u , we can store the assigned cell for u in the satellite data of u . The query is then performed on \mathcal{H} , using σ and τ as the query vertices. When a site s is inserted into or deleted in S , only the edges incident to the assigned cell σ are affected. By Lemma 3.1, there are only $O(1)$ such edges. Thus, once the set E of these edges is determined, by Theorem 2.1, we can update \mathcal{H} in time $O(\log^2 n)$.

It remains to find the edges E of H that change when we update S . For each pair σ, τ of neighboring cells in \mathcal{G}_1 , we maintain a maximal bichromatic matching (MBM) $M_{\{\sigma, \tau\}}$ for $R = S(\sigma)$ and $B = S(\tau)$, as in Lemma 2.3 (note that the special requirements of the lemma are met in our case). By construction, there is an edge between σ and τ in H if and only if $M_{\{\sigma, \tau\}}$ is not empty. When inserting or deleting a site s from S , we proceed as follows: let $\sigma \in \mathcal{G}_1$ be the cell associated to σ . We go through all cells $\tau \in N_{5 \times 5}(\sigma)$, and we update $M_{\{\sigma, \tau\}}$ by inserting or deleting s from the relevant set. If $M_{\{\sigma, \tau\}}$ becomes non-empty during an insertion or empty during a deletion, we add the edge $\sigma\tau$ to E and mark it for insertion or deletion, respectively. Putting everything together, we obtain the main result of this section:

► **Theorem 3.2.** *There is a dynamic connectivity structure for unit disk graphs such that an update takes amortized time $O(\log^2 n)$ and a connectivity query takes worst-case time $O(\log n / \log \log n)$, where n is the maximum number of sites. The structure uses $O(n)$ space.*

4 Fully dynamic bounded radius ratio

We extend our structure from Theorem 3.2 to the case of bounded radius ratio Ψ . Now, the running times will depend polynomially on Ψ . The general approach is unchanged, but the varying sizes of the disks introduce new issues. First, we adapt Theorem 3.2 to disks of different sizes. Instead of just \mathcal{G}_1 , we rely on a hierarchical grid with $\lceil \log \Psi \rceil + 1$ levels. Each site s is assigned to a cell σ of such level that $|\sigma| \leq r_s < 2|\sigma|$. Since the disks have different sizes, we can no longer use Lemma 2.3 to maintain the maximal bichromatic matchings (MBMs) between neighboring non-empty grid cells. Instead, we use the more complex structure from Lemma 2.2. This increases the overhead for updating the MBM for each pair of neighboring cells. Furthermore, a disk can now intersect disks from $\Theta(\Psi^2)$ other cells, instead of the $O(1)$ -bound from the unit disk case, see Figure 4. Thus, the degree of the proxy graph and the number of edges that need to be modified in a single update becomes much larger. This results in the following theorem, see the full version for details.

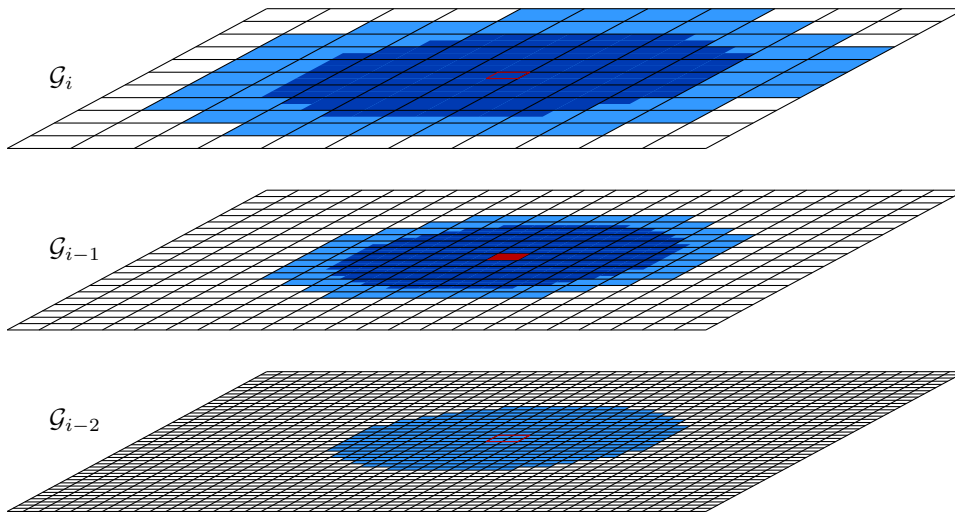
► **Theorem 4.1.** *There is a dynamic connectivity structure for disk graphs of bounded radius ratio Ψ such that an update takes amortized expected time $O(\Psi^2 \lambda_6(\log n) \log^7 n)$ and a connectivity query takes worst-case time $O(\log n / \log \log n)$, where n is the maximum number of sites at any time. The data structure requires $O(\Psi^2 n \log n)$ expected space.*

To remedy this latter problem – at least partially – we describe in Section 4.1 how to refine the proxy graph so that fewer edges need to be modified in a single update operation. This will reduce the dependence on Ψ in the update time to linear. The query procedure becomes slightly more complicated, but the asymptotic running time remains unchanged.

Note that the approach described above is similar to the method of Kaplan et al. [9, Theorem 9.11] that achieves the same time and space bounds. However, the details of our implementation are crucial for the adaptation in Section 4.1. Most significantly, our implementation uses a hierarchical grid instead of a single fine grid.

4.1 Improving the dependence on Ψ

To avoid an update time dependent on the potentially quadratic number of neighbors, we show how to reduce the degree of the proxy graph H from $\Theta(\Psi^2)$ to $O(\Psi)$. The intuition is that to maintain the connected components of $\mathcal{D}(S)$, it suffices to focus on *maximal* disks that are not contained in any other disk in S . From this, it follows that we only need to consider edges between disks that intersect properly. When we want to perform a connectivity query between sites s and t , we must find appropriate maximal disks that contain s and t . Let D



■ **Figure 4** The neighborhood of the red colored cell in \mathcal{G}_{i-1} . The area of the neighboring cells in one level beneath is colored in a darker shade.

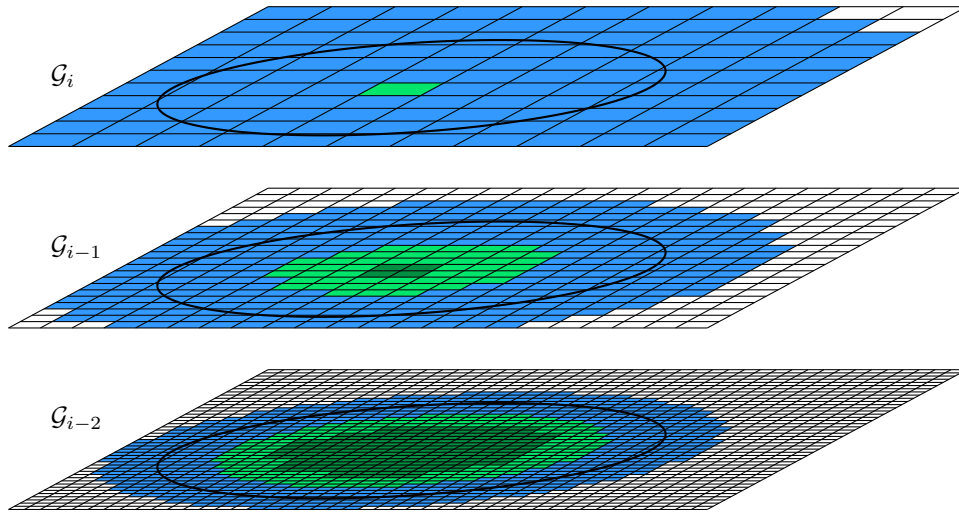
be a disk and $\sigma \in \mathcal{G}$ a cell. We say that σ is *fully covered* by D if and only if every possible assigned disk of σ is fully contained in D . We call σ *maximal* if and only if there is no larger cell $\tau \supset \sigma$ that is fully covered by D_s .

Given a disk D , the maximal cells in the quadforest \mathcal{F} that are fully covered by D are exactly those that are closest to the root in their quadtrees. Furthermore, the whole subtree of \mathcal{F} that is rooted in a maximal fully covered cell consists of cells that are fully covered by D . The following lemma bounds the number of the different types of cells. See Figure 5.

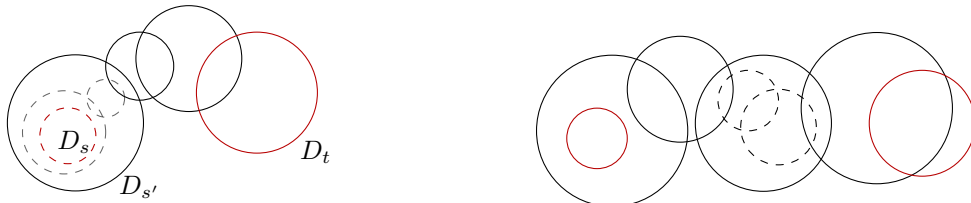
► **Lemma 4.2.** *Let $s \in S$ be a site, and let \mathcal{N} be the cells of \mathcal{F} that may contain a disk that intersects D_s . Write $\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2 \cup \mathcal{N}_3$, where \mathcal{N}_1 are the cells that are not fully covered by D_s , \mathcal{N}_2 the disks that are maximal fully covered by D_s , and \mathcal{N}_3 the disks that are fully covered by D_s , but not maximal with this property. Then, we have $|\mathcal{N}_1 \cup \mathcal{N}_2| = O(\Psi)$ and $|\mathcal{N}_3| = O(\Psi^2)$. Using the quadforest \mathcal{F} , we can find $\mathcal{N}_1 \cup \mathcal{N}_2$ in $O(\Psi + \log n)$ time and \mathcal{N}_3 in $O(\Psi^2 + \log n)$ time.*

Proof sketch. (Full proof in the full version) The cells of \mathcal{N}_1 form an annulus per level. A volume argument shows that they sum up to $O(\Psi)$ altogether. Now, note that every cell in \mathcal{N}_2 is either a quadtree root in \mathcal{F} or a child of a cell in \mathcal{N}_1 . Hence, we have $|\mathcal{N}_1 \cup \mathcal{N}_2| = O(\Psi)$. The bound on $|\mathcal{N}_3|$ follows from the number of neighbors. The retrieval is possible with simple traversal after finding the relevant roots of \mathcal{F} in $O(\log n)$ time. ◀

Now, we show that it is enough to focus on a subset of the edges in the proxy graph. More precisely, let H' be the subgraph of H that is defined as follows: as in H , the vertices of H' are all cells σ that have $S(\sigma) \neq \emptyset$. Two cells σ, τ with $|\sigma| \geq |\tau|$ are adjacent in H' if and only if there are $s \in S(\sigma)$ and $t \in S(\tau)$ such that D_s and D_t intersect and such that D_s does not fully cover τ . Let σ be a cell in H . We define the *proxy cell* σ' of σ as follows: if there is no disk in S that fully covers σ , then $\sigma' = \sigma$. Otherwise, let $\bar{\sigma} \supseteq \sigma$ be the maximal cell that contains σ and is fully covered by a disk in S , and let $D_s, s \in S$, be a disk of maximum radius that fully covers $\bar{\sigma}$. Then, we set σ' to be the cell with $s \in S(\sigma')$. If there are multiple such disks, the choice is arbitrary.



■ **Figure 5** The types of cells that require checking when updating the black disk in Theorem 4.1:
 ■ \mathcal{N}_1 : not fully covered ■ \mathcal{N}_2 : maximal fully covered ■ \mathcal{N}_3 : fully covered, not maximal.



(a) Omitting the dashed disks and querying for s' instead of s still leads to a valid path to t . (b) A path between the two red disks can ignore the dashed black disks as intermediates.

■ **Figure 6** Depiction of the arguments in Lemma 4.3.

► **Lemma 4.3.** *Let $s, t \in S$ be two sites, and let σ, τ be the cells with $s \in S(\sigma)$ and $t \in S(\tau)$. Let σ' and τ' be the proxy cells for σ and τ . Then, σ' and τ' are connected in H' if and only if s and t are connected in $\mathcal{D}(S)$.*

Proof sketch. (Full proof in the full version) First, suppose that s and t are not connected in $\mathcal{D}(S)$. Since H' is a subgraph of H , it follows that σ' and τ' are not connected in H' .

Next, suppose that s and t are connected in $\mathcal{D}(S)$. We consider a path of (inclusion) maximal disks that connects s and t in $\mathcal{D}(S)$, and we show that it induces a path between σ' and τ' in H' . Let $D_{s'}, D_{t'}$ with $s' \in S(\sigma')$, $t' \in S(\tau')$ be the disks of maximum radius which caused σ', τ' to be proxy cells of σ, τ . Now, there is a path π in $\mathcal{D}(S)$ between s' and t' that uses only maximal disks: indeed, along any path in $\mathcal{D}(S)$ between s' and t' , we can replace every disk by a maximal disk that contains it, and the resulting path π (possibly after removing duplicate disks) has the required property. See Figure 6. Consider the sequence π' of cells in H' that we obtain by replacing every site u in π by the cell σ_u in H' with $u \in S(\sigma_u)$, and by removing any duplicate cells. We observe that π' is actually a path in H' , since the assigned cells for two intersecting maximal disks of S must be adjacent in H' . ◀

Now, our strategy is to maintain the proxy graph H' instead of the graph H , again such that each potential edge of H' is supported by an MBM structure. This will make the updates faster. However, when performing a query, we must be able to find the proxy cells for the query sites efficiently. This requires a further modification of the quadforest \mathcal{F} .

► **Theorem 4.4.** *There is a data structure for dynamic disk connectivity with expected amortized update time $O(\Psi \lambda_6(\log n) \log^7 n)$ and amortized query time $O(\log n / \log \log n)$. It needs $O(\Psi n \log n)$ expected space.*

Proof sketch. (Full proof in the full version) We may assume that $\Psi = O(n^3)$. We augment the quadforest \mathcal{F} : in each cell σ in \mathcal{F} , we store the set \mathcal{C}_σ of all sites $s \in S$ such that σ is maximal fully covered by D_s . \mathcal{C}_σ is organized as a max-heap, ordered by radius r_s .

We describe how to insert a new site s . First, we insert s into the quadforest \mathcal{F} . Then, we obtain the sets \mathcal{N}_1 and \mathcal{N}_2 for s using Lemma 4.2 and insert them into \mathcal{F} . For each $\tau \in \mathcal{N}_1$ we insert s into the MBM for σ and τ and update \mathcal{H} of Theorem 2.1 accordingly. For each $\tau \in \mathcal{N}_2$, we insert s into the max-heap \mathcal{C}_τ . A deletion is handled analogously.

To perform a connectivity query between s and t , let σ and τ be the cells with $s \in S(\sigma)$ and $t \in S(\tau)$. We determine the proxy cell σ' via obtaining the maximal cell $\bar{\sigma} \supseteq \sigma$ that contains σ and is fully covered by a disk from S . Let u be the site of maximum radius in the max-heap $\mathcal{C}_{\bar{\sigma}}$ and set $\sigma' = \sigma_u$. τ' is obtained similarly. Afterwards, \mathcal{H} is queried with σ' and τ' for the final result. By Lemma 4.3, this gives the correct answer. The overall running time for this query procedure is $O(\log n)$, where the bottleneck consists in ascending the quadtree.

The query time can be improved via maintaining every $\Theta(\log \log \Psi)$ levels shortcuts pointing upwards, each pointing to the next. To decide whether to take a shortcut, the respective cells have another max-heap containing all intermediate max-heaps. ◀

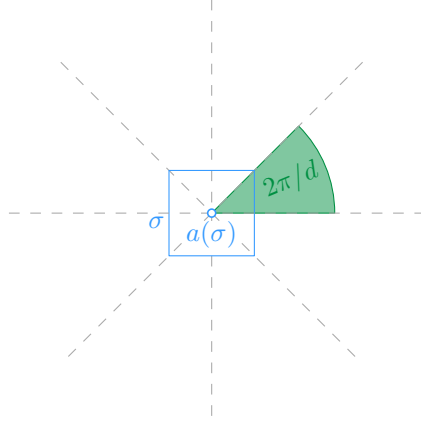
5 Semi-dynamic bounded radius ratio

We turn to the semi-dynamic setting, and we show how to reduce the dependency on Ψ from linear to logarithmic. For both the incremental and the decremental scenario, we use the same proxy graph H to represent the connectivity in $\mathcal{D}(S)$. The proxy graph is described in Section 5.1. In Section 5.2 we then describe the data structures using H . For details on how to use the proxy graph in the incremental setting, refer to the full version of this paper.

5.1 The proxy graph

The vertex set of the proxy graph H contains one vertex for each site in S , plus one additional vertex per certain region $A \in \mathcal{A}$ in the plane, to be described below. Each region is defined based on a cell of a quadtree and associated with two site sets, $S_1(A)$ and $S_2(A)$. The first set $S_1(A) \subseteq S$ is defined such that all sites $s \in S_1(A)$ lie in A and have a radius r_s comparable to the size of A , for a notion of “comparable” to be detailed below. A site s can lie in several sets $S_1(A)$. We will ensure that for each region A , the induced disk graph $\mathcal{D}(S_1(A))$ of the associated sites is a clique. The second set $S_2(A) \subseteq S$ contains a site s if it lies in the cell associated to the region A and if r_s is “small”. The sites in $S_2(A)$ are all sites with a suitable radius in the associated cell of A that have an edge in $\mathcal{D}(S)$ to at least one site in $S_1(A)$.

The proxy graph H is bipartite, with all edges going between the *site-vertices* and the *region-vertices*. The edges of H connect every region A to all sites in $S_1(A) \cup S_2(A)$. The connections between the sites in $S_1(A)$ and A constitute a sparse representation of the corresponding clique $\mathcal{D}(S_1(A))$. The edges connecting a site in $S_2(A)$ to A allow us to represent all edges in $\mathcal{D}(S)$ between $S_2(A)$ and $S_1(A)$ by two edges in H , and since $\mathcal{D}(S_1(A))$ is a clique, this sparse representation does not change the connectivity between the sites. We will see that the sites in $S_2(A)$ can be chosen such that every edge in $\mathcal{D}(S)$ is represented by two edges in H . Furthermore, we will ensure that the number of regions, and the total size of the associated sets $S_1(A)$ and $S_2(A)$ is small, giving a sparse proxy graph.



■ **Figure 7** The cones \mathcal{C}_d with angle $2\pi/d$, with apex at the center $a(\sigma)$ of a cell σ .

Now, we describe the details of the regions in \mathcal{A} . For each site $s \in S$ we consider the cell $\sigma_s \in \mathcal{G}$ with $s \in \sigma_s$ and $|\sigma_s| \leq r_s < 2|\sigma_s|$ and its (15×15) -neighborhood $N(s)$. We let $\mathcal{N} = \{N(s) \mid s \in S\}$ and construct the quadforest \mathcal{F} for \mathcal{N} . This quadforest \mathcal{F} contains quadtrees that cover the lowest $\lfloor \log \Psi \rfloor + 1$ levels of the hierarchical grid \mathcal{G} , see the full version for details. The set \mathcal{A} of region-vertices of H is a subset of the set $\mathcal{A}_{\mathcal{F}}$ that contains certain regions for every cell of \mathcal{F} . There are three kinds of regions for a cell σ of \mathcal{F} : the *outer regions*, the *middle regions*, and the *inner region*.

To describe these regions, we first define for $d \in \mathbb{N}$ a set \mathcal{C}_d of d cones with opening angle $2\pi/d$, such that all cones in \mathcal{C}_d have their apex in the origin, have pairwise disjoint interiors, and cover the plane. For a cell $\sigma \in \mathcal{F}$, we denote by $\mathcal{C}_d(\sigma)$ a translated copy of \mathcal{C}_d whose apex has been moved to the center $a(\sigma)$, of σ , as shown in Figure 7.

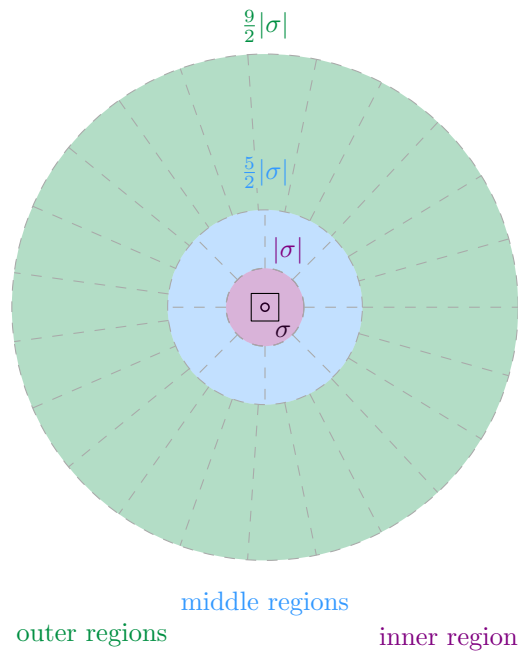
Let $\Gamma(a, r_1, r_2)$ be the annulus centered at a with inner radius r_1 and outer radius r_2 . To define the outer regions for a cell σ , we consider the set $\mathcal{C}_{d_1}(\sigma)$, for some integer parameter d_1 to be determined below. For each \mathcal{C}_{d_1} we set the outer regions to be $\{C \cap \Gamma(a(\sigma), \frac{5}{2}|\sigma|, \frac{9}{2}|\sigma|) \mid C \in \mathcal{C}_{d_1}\}$. Similarly to this, we define the middle regions as $\{C \cap \Gamma(a(\sigma), |\sigma|, \frac{5}{2}|\sigma|) \mid C \in \mathcal{C}_{d_2}\}$. Finally, the inner region for σ is the disk with center $a(\sigma)$ and radius $|\sigma|$. See Figure 8 for an illustration of the regions for a cell σ .

We associate a set of sites $S_1(A) \subseteq S$ with each region $A \in \mathcal{A}_{\mathcal{F}}$. The set $S_1(A)$ contains all sites t such that (i) $t \in A$; (ii) $|\sigma| \leq r_t < 2|\sigma|$; and (iii) $\|a(\sigma)t\| \leq r_t + \frac{5}{2}|\sigma|$. This means that the disk D_t has size comparable to $|\sigma|$, a center in A . If t is in a middle or inner region, the third property is trivially true. If t is in an outer region it implies that t intersects the inner boundary of A .

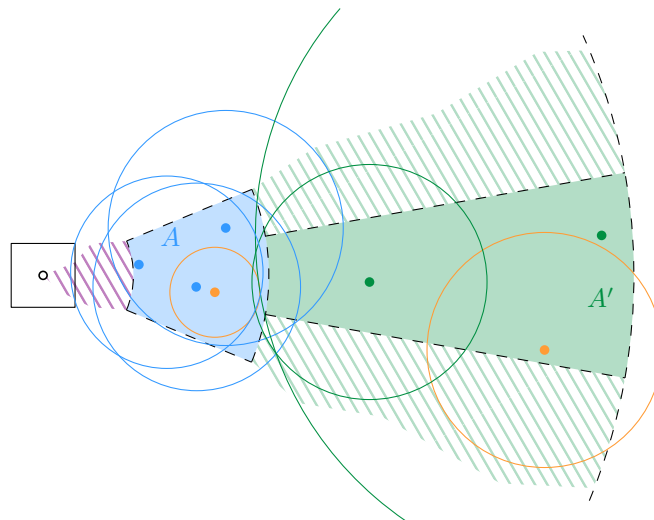
We define $\mathcal{A} \subseteq \mathcal{A}_{\mathcal{F}}$ as the set of regions where $S_1(A) \neq \emptyset$. In the following, we will not strictly distinguish between a vertex from \mathcal{A} and the corresponding region, provided it is clear from the context.

For each region $A \in \mathcal{A}$, we define a set $S_2(A)$ as the set of all sites s such that (i) $s \in \sigma$; (ii) s is adjacent in $\mathcal{D}(S)$ to at least one site in $S_1(A)$; and (iii) $r_s < 2|\sigma|$.

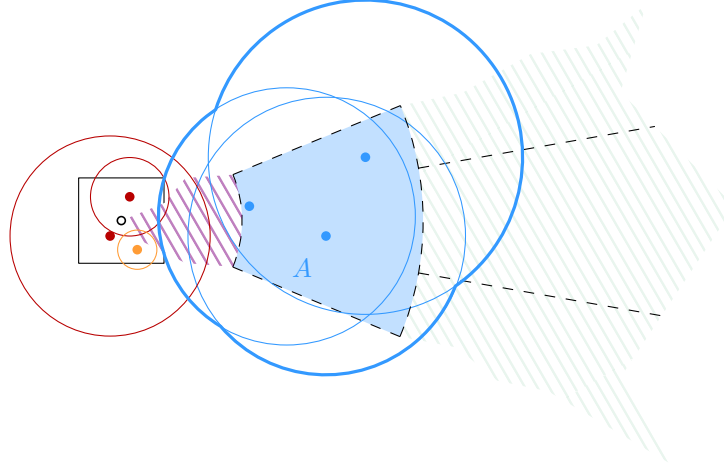
We add an edge sA in H between a site s and a region A if and only if $s \in S_1(A) \cup S_2(A)$. Note that the sets $S_1(A)$ and $S_2(A)$ are not necessarily disjoint, as for the center region defined by a cell σ , a site with $|\sigma| \leq r_s < 2|\sigma|$ will be both in $S_1(A)$ and $S_2(A)$. However, this will adversely affect neither the preprocessing time nor the correctness. The following structural lemma will help us both to show that H accurately represents the connectivity as well as to bound the size of H and the preprocessing time in the decremental setting.



■ **Figure 8** The regions defined by a cell σ .



■ **Figure 9** The set $S_1(A)$ is marked blue. The orange site in A is not in the set because its radius is too small. The orange site in A' is not in $S_1(A')$: even though its radius is in the correct range, it does not touch or intersect the inner boundary.



■ **Figure 10** The red sites in σ are in $S_2(A)$. The radius of the orange site is in the correct range, but it does not intersect a site in $S_1(A)$ (marked blue).

► **Lemma 5.1.** *Let st be an edge in $\mathcal{D}(S)$ with $r_s \leq r_t$, then*

1. *there is a cell $\sigma \in N(t)$ with $s \in \sigma$ such that σ defines a region A with $t \in A$; and*
2. *all cells that define a region A with $t \in S_1(A)$ are in $N(t)$.*

The proof for Lemma 5.1 can be found in the full version of the paper. Before we argue that H accurately represents the connectivity of $\mathcal{D}(S)$, we show that the associated sites of a region in \mathcal{A} form a clique in $\mathcal{D}(S)$.

► **Lemma 5.2.** *Suppose that $d_1 \geq 23$ and $d_2 \geq 8$. Then, for any region $A \in \mathcal{A}$, the associated sites in $S_1(A)$ form a clique in $\mathcal{D}(S)$.*

Proof sketch. (Full proof in the full version) The diameter of the inner and middle regions is at most $2|\sigma|$, thus two sites in $S_1(A)$ always intersect.

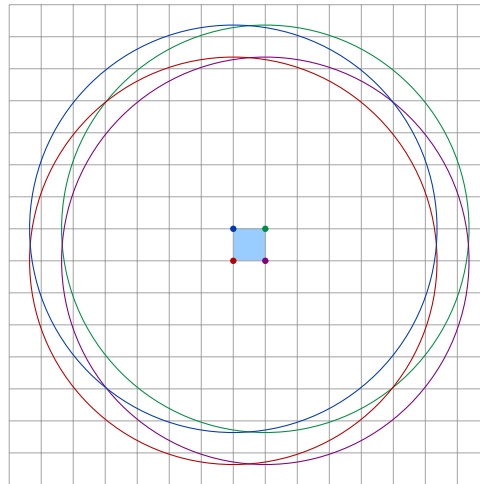
If a site t lies in the outer region, we can show that the line segments that are perpendicular to the boundary rays of the cones, go through t and are inside the cone are contained in D_t . Then any other site t' that has a larger distance to $a(\sigma)$ than t either lies in the convex hull defined by the perpendicular line segments, or $D_{t'}$ contains a line segment that intersects the convex hull, see Figure 11. ◀

Having Lemmas 5.1 and 5.2 at hand, we can now show that H accurately represents the connectivity of $\mathcal{D}(S)$.

► **Lemma 5.3.** *Two sites are connected in H if and only if they are connected in $\mathcal{D}(S)$.*

Proof. Let $s, t \in S$. First, we show that if s and t are connected in H , they are also connected in $\mathcal{D}(S)$. The path between s and t in H alternates between vertices in S and vertices in \mathcal{A} . Thus, it suffices to show that if two sites u and u' are connected with the same region $A \in \mathcal{A}$, they are also connected in $\mathcal{D}(S)$. This follows directly from Lemma 5.2: if u and u' both lie in $S_1(A)$, they are part of the same clique. Otherwise, $S_2(A)$ is non-empty, and there is at least one site in $S_1(A)$ which intersects the site in $S_2(A)$. Then u is connected to u' via the clique induced by $S_1(A)$, and the claim follows.

Now, we consider two sites connected in $\mathcal{D}(S)$, and we show that they are also connected in H . It suffices to show that if s, t are adjacent in $\mathcal{D}(S)$, they are connected in H . Assume without loss of generality that $r_s \leq r_t$, and let σ be the cell in $N(t)$ with $s \in \sigma$. The cell



■ **Figure 11** The disk $D(t, \frac{9}{2}|\sigma|)$ is contained in $N_{15 \times 15}(\tau)$.

σ exists by the first property of Lemma 5.1, and it belongs to \mathcal{F} , since σ lies in the first $\lceil \log \Psi \rceil + 1$ levels of \mathcal{G} and since $\sigma_s \subseteq \sigma$. Thus, we get that $t \in S_1(A)$ for some $A \in \mathcal{A}_{\mathcal{F}}$. As the regions with non-empty sets $S_1(A)$ are in \mathcal{A} , by definition, the edge tA exists in H .

Now we argue that $s \in S_2(A)$, and thus the edge As also exists in H . This follows by straightforward checking of the properties of a site in $S_2(A)$. We have $s \in \sigma$ by the definition of σ , and, by assumption, $r_s \leq r_t < 2|\sigma|$. Finally, as t is in $S_1(A)$ and as D_s and D_t intersect, there is at least one site in $S_1(A)$ that intersects D_s . The claim follows. ◀

After we have shown that H accurately represents the connectivity relation in $\mathcal{D}(S)$, we now show that the number of edge in H depends only on n and Ψ , and not on the number of edges in $\mathcal{D}(S)$ or the diameter of S . The proof of the following lemma can be found in the full version.

► **Lemma 5.4.** *The proxy graph H has $O(n)$ vertices and $O(n \log \Psi)$ edges.*

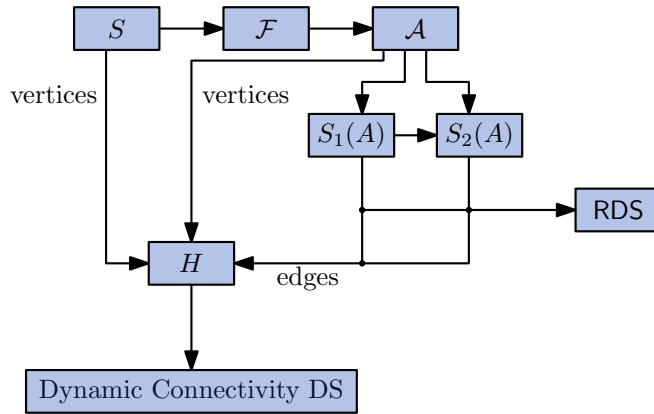
5.2 The decremental data structure

The decremental data structure has several components: we store a quadforest containing the cells defining \mathcal{A} and for every $A \in \mathcal{A}$, we store the sets $S_1(A)$ and $S_2(A)$. For each region $A \in \mathcal{A}$, we store a disk revealing structure (RDS) as in Theorem 1.1 with $B = S_1(A)$ and $R = S_2(A)$. Finally, we store the proxy graph H in a Holm et al. data structure \mathcal{H} [8]. See Figure 12 for an illustration.

As usual, the connectivity queries are answered using \mathcal{H} . To delete a site s , we first remove from \mathcal{H} all incident edges of s . Then, we go through all regions A with $s \in S_1(A)$. We remove s from $S_1(A)$ and the RDS of A , and we let U be the set of revealed sites from $S_2(A)$ reported by the RDS. We delete each such site $u \in U$ from $S_2(A)$ and the corresponding RDS. Additionally, we delete the edges uA for $u \in U$ from \mathcal{H} for all $u \in U$ that are not also in $S_1(A)$. Next, for each region A with $s \in S_2(A)$, we remove s from $S_2(A)$ and the associated RDS.

This gives us a time bound for the preprocessing time and the main theorem follows.

► **Lemma 5.5.** *Given a set S of n sites, we can construct the data structure described above in $O(n \log^5 n \lambda_6(\log n) + n \log \Psi \log^3 n)$ time.*



■ **Figure 12** The structure of the decremental data structure.

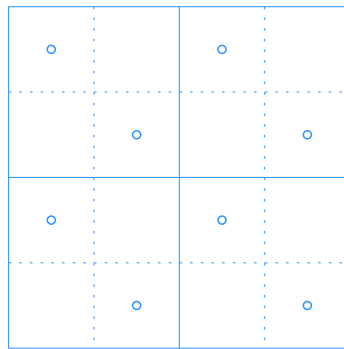
► **Theorem 5.6.** *The data structure handles m site deletions in overall $O((n \log^5 n + m \log^7 n) \lambda_6(\log n) + n \log \Psi \log^4 n)$ time. Furthermore, it correctly answers connectivity queries in $O(\log n / \log \log n)$ amortized time.*

6 Semi-dynamic arbitrary radius ratio

We extend the approach from Section 5 to obtain a decremental data structure with a running time that is independent of Ψ . The cost for dropping the dependence on Ψ is replacing the additive $O(n \log \Psi \log^4 n)$ term in the running time of Theorem 5.6 with an additional $O(\log n)$ factor in the first term. The $O(n \log \Psi \log^4 n)$ term in Theorem 5.6 arose from the total size of the sets $S_2(A)$, and thus from the height of the quadtrees in \mathcal{F} . We can get rid of this dependency by using a *compressed quadtree* \mathcal{Q} instead of \mathcal{F} . The height and size of \mathcal{Q} do not depend on the radius ratio of the diameter of S , but only on n . Nonetheless, the height of \mathcal{Q} could still be $\Theta(n)$, which is not favorable for our purposes. In order to reduce the number of edges in our proxy graph to $O(n \log n)$, we use a *heavy path decomposition* of \mathcal{Q} in combination with a *canonical decomposition* for every heavy path. Let $\text{diam}(S) = \max_{s,t \in S} \|st\|$. To simplify our arguments, we assume without loss of generality that S and its associated radii are scaled all associated radii are at least 1. This allows us to keep working with our hierarchical grid \mathcal{G} , as defined in Section 2.

Compressed quadtrees. The quadtree defined for a set \mathcal{C} of $O(n)$ cells as in Section 2, has $O(n)$ leaves and height $O(\log(|\rho|))$, where ρ is the smallest cell in \mathcal{G} that contains all cells of \mathcal{C} . This height can be arbitrarily large, even if n is small. To avoid this, we use the notion of a *compressed quadtree* \mathcal{Q} as defined by Har-Peled [7] among others. \mathcal{Q} has $O(n)$ vertices, height $O(n)$, and it can be constructed in $O(n \log n)$ time [2, 7]. While the latter construction algorithm is stated for planar point sets it can be applied by considering a set of $O(n)$ virtual sites, similar to a construction of Har-Peled [7], see Figure 13.

Heavy paths. Let T be a rooted ordered tree. An edge $uv \in T$ is called *heavy* if v is the first child of u that maximizes the total number of nodes in the subtree rooted at v . Otherwise, the edge uv is *light*. By definition, every interior node in T has exactly one child that is connected by a heavy edge. A *heavy path* is a maximum path in T that consists only of heavy edges. The *heavy path decomposition* of T is the set of all the heavy paths in T . The following lemma summarizes a classic result on the properties of heavy path decompositions.



■ **Figure 13** Four cells from $N_{15 \times 15}(\sigma)$ with the virtual sites.

► **Lemma 6.1** (Sleator and Tarjan [12]). *Let T be a tree with n vertices. Then, the following properties hold:*

1. *Every leaf-root path in T contains $O(\log n)$ light edges;*
2. *every vertex of T lies on exactly one heavy path; and*
3. *the heavy path decomposition of T can be constructed in $O(n)$ time.*

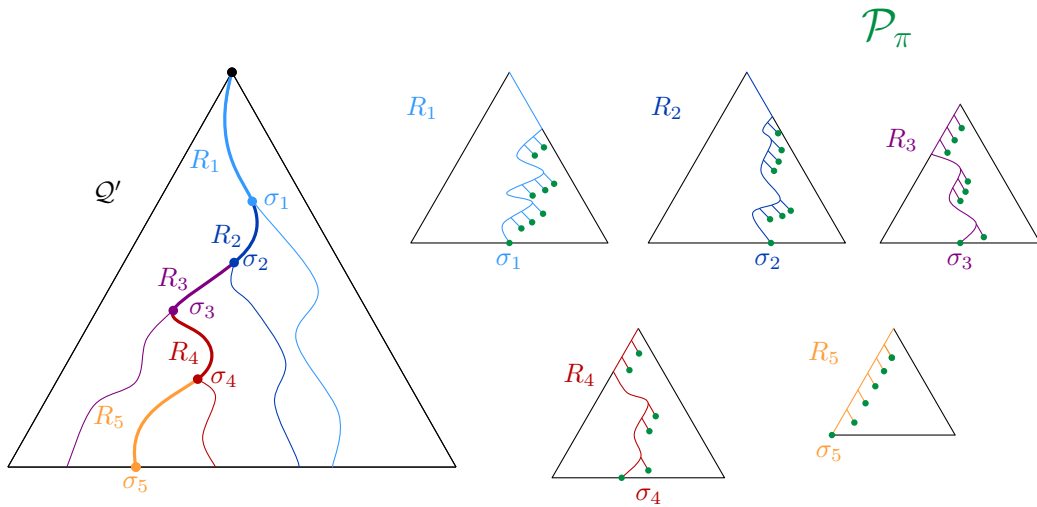
The proxy graph. The general structure of the proxy graph is as in Section 5.1, and we will often refer back to it. We still have a bipartite graph with S on one side and a set of regions vertices on the other side. The regions will again be used to define sets $S_1(A)$ and $S_2(A)$ that will determine the edges. However, we will adapt the regions A and define them based certain *subpaths* of the compressed quadtree \mathcal{Q} instead of single cells. Furthermore, we will relax the condition on the radii in the definition of the sets $S_1(A)$.

As usual, for a site $s \in S$, let σ_s be the cell in \mathcal{G} with $s \in \sigma_s$ and $|\sigma_s| \leq r_s < 2|\sigma_s|$. Let $N(s)$ be the (15×15) -neighborhood of σ_s . Let $\mathcal{N} = \{N(s) \mid s \in S\}$, and let \mathcal{Q} be the compressed quadtree for \mathcal{N} . Now, let \mathcal{R} be the heavy path decomposition of \mathcal{Q} , as in Lemma 6.1. For each heavy path $R \in \mathcal{R}$, we find a set \mathcal{P}_R of *canonical paths* such that every subpath of R can be written as the disjoint union of $O(\log n)$ canonical paths. To be precise, for each $R \in \mathcal{R}$, we build a *biased* binary search tree T_R with the cells of R in the leaves, sorted by increasing diameter. The weights in the biased binary search tree are chosen as described by Sleator and Tarjan [12]: for a node σ of R , let the weight w_σ be the number of nodes in \mathcal{Q} that are below σ (including σ), but not below another node of R below σ . Then, the depth of the leaf σ in T_R is $O(\log(w_R/w_\sigma))$, where w_R is the total weight of all leaves in T_R . We associate each vertex v in T_R with the path induced by the cells in the subtree rooted at v , and we add this path to \mathcal{P}_R . Using this construction, we can write every path in \mathcal{Q} that starts at the root as the disjoint union of $O(\log n)$ canonical path:

► **Lemma 6.2.** *Let σ be a vertex of \mathcal{Q} , and let π be the path from the root of \mathcal{Q} to σ . There exists a set \mathcal{P}_π of canonical paths such that: (i) $|\mathcal{P}_\pi| = O(\log n)$; and (ii) π is the disjoint union of the canonical paths in \mathcal{P}_π .*

Proof sketch. (Full proof in full version) By Lemma 6.1 there are $O(\log n)$ heavy paths R_1, \dots, R_k along π . The subpaths defined by the search paths to the smallest cell of a heavy path R_i partition π . Furthermore, by summing over the weights of the leaves of the biased binary search tree, we get that the overall number of canonical paths for π is $O(\log n)$. ◀

The vertex set of the proxy graph H again consists of S and a set of regions \mathcal{A} . We define $O(1)$ regions for each canonical path R in a similar way as in Section 5.1. Let σ be the smallest cell and τ the largest cell of R . The *inner* and *middle regions* of R are defined



■ **Figure 14** Illustration of Lemma 6.2. On the left, we see the decomposition of R into R_1, \dots, R_k . On the right, the vertices defining \mathcal{P}_π are depicted in green.

as in Section 5.1, using σ as the defining cell. For the *outer regions* of R , we extend the outer radius of the annulus: they are defined as the intersections of the cones in \mathcal{C}_{d_1} with the annulus of inner radius $\frac{5}{2}|\sigma|$ and outer radius $\frac{5}{2}|\sigma| + 2|\tau|$, again centered at $a(\sigma)$. The set \mathcal{A} now contains the regions defined in this way for all canonical paths.

Given a region $A \in \mathcal{A}$ for a canonical path R with smallest cell σ and largest cell τ , we can now define the sets $S_1(A)$ and $S_2(A)$. These definitions are similar to the analogous sets in Section 5.1. The set $S_1(A)$ contains all sites t such that (i) $t \in A$; (ii) $|\sigma| \leq r_t \leq 2|\tau|$; and (iii) $\|a(\sigma)t\| \leq r_t + \frac{5}{2}|\sigma|$. The definition for $S_2(A)$ is also similar to Section 5.1, using canonical paths instead of cells. Let $s \in S$ be a site, and π_s be the path in \mathcal{Q} from the root to σ_s . Let \mathcal{P}_s be the decomposition of π_s into canonical paths as in Lemma 6.2. Let A be a region, defined by a canonical path P . Then, $s \in S_2(A)$ if (i) $P \in \mathcal{P}_s$; and (ii) s is adjacent in $\mathcal{D}(S)$ to at least one site in $S_1(A)$. These are basically the conditions we had in Section 5.1. However, as the definition is restricted to those canonical paths in \mathcal{P}_{π_s} , not all sites satisfying these conditions are considered. Using similar arguments as in Section 5.1, this suffices to make sure that the proxy graph represents the connectivity, while also ensuring that each site s lies in few sets $S_2(A)$.

The graph H is now again defined by connecting each region $A \in \mathcal{A}$ to all sites in $s \in S_1(A) \cup S_2(A)$. By similar considerations as in Section 5, we obtain a decremental data structure for disk graphs with arbitrary radii. The details can be found in the full version.

References


- 1 Pankaj K. Agarwal, Ravid Cohen, Dan Halperin, and Wolfgang Mulzer. Maintaining the union of unit discs under insertions with near-optimal overhead. In *Proc. 35th Annu. Sympos. Comput. Geom. (SoCG)*, pages 26:1–26:15, 2019. doi:10.4230/LIPIcs.SoCG.2019.26.
- 2 Kevin Buchin, Maarten Löffler, Pat Morin, and Wolfgang Mulzer. Preprocessing imprecise points for Delaunay triangulation: Simplified and extended. *Algorithmica*, 61(3):674–693, 2011. doi:10.1007/s00453-010-9430-0.
- 3 Timothy M. Chan. Dynamic geometric data structures via shallow cuttings. *Discrete Comput. Geom.*, 64(4):1235–1252, 2020. doi:10.1007/s00454-020-00229-5.

- 4 Timothy M. Chan, Mihai Pătraşcu, and Liam Roditty. Dynamic connectivity: Connecting to networks and geometry. *SIAM J. Comput.*, 40(2):333–349, 2011. doi:10.1137/090751670.
- 5 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009.
- 6 David Eppstein, Giuseppe F Italiano, Roberto Tamassia, Robert E Tarjan, Jeffery Westbrook, and Moti Yung. Maintenance of a minimum spanning forest in a dynamic plane graph. *J. Algorithms*, 13(1):33–54, 1992. doi:10.1016/0196-6774(92)90004-V.
- 7 Sarel Har-Peled. *Geometric Approximation Algorithms*, volume 173. American Mathematical Society, 2011. doi:10.1090/surv/173.
- 8 Jacob Holm, Kristian de Lichtenberg, and Mikkel Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *J. ACM*, 48(4):723–760, 2001. doi:10.1145/502090.502095.
- 9 Haim Kaplan, Wolfgang Mulzer, Liam Roditty, Paul Seiferth, and Micha Sharir. Dynamic planar Voronoi diagrams for general distance functions and their algorithmic applications. *Discrete Comput. Geom.*, 64(3):838–904, 2020. doi:10.1007/s00454-020-00243-7.
- 10 Chih-Hung Liu. Nearly optimal planar k nearest neighbors queries under general distance functions. In *Proc. 31st Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 2842–2859, 2020. doi:10.1137/1.9781611975994.173.
- 11 Micha Sharir and Pankaj K. Agarwal. *Davenport-Schinzel sequences and their geometric applications*. Cambridge University Press, 1995.
- 12 Daniel D. Sleator and Robert Endre Tarjan. A data structure for dynamic trees. *J. Comput. System Sci.*, 26(3):362–391, 1983. doi:10.1016/0022-0000(83)90006-5.
- 13 Mikkel Thorup. Near-optimal fully-dynamic graph connectivity. In *Proc. 32nd Annu. ACM Sympos. Theory Comput. (STOC)*, pages 343–350, 2000.

An $(\aleph_0, k + 2)$ -Theorem for k -Transversals

Chaya Keller  

Ariel University, Israel

Micha A. Perles 

Einstein Institute of Mathematics, Hebrew University, Jerusalem, Israel

Abstract

A family \mathcal{F} of sets satisfies the (p, q) -property if among every p members of \mathcal{F} , some q can be pierced by a single point. The celebrated (p, q) -theorem of Alon and Kleitman asserts that for any $p \geq q \geq d + 1$, any family \mathcal{F} of compact convex sets in \mathbb{R}^d that satisfies the (p, q) -property can be pierced by a finite number $c(p, q, d)$ of points. A similar theorem with respect to piercing by $(d - 1)$ -dimensional flats, called $(d - 1)$ -transversals, was obtained by Alon and Kalai.

In this paper we prove the following result, which can be viewed as an $(\aleph_0, k + 2)$ -theorem with respect to k -transversals: Let \mathcal{F} be an infinite family of sets in \mathbb{R}^d such that each $A \in \mathcal{F}$ contains a ball of radius r and is contained in a ball of radius R , and let $0 \leq k < d$. If among every \aleph_0 elements of \mathcal{F} , some $k + 2$ can be pierced by a k -dimensional flat, then \mathcal{F} can be pierced by a finite number of k -dimensional flats.

This is the first (p, q) -theorem in which the assumption is weakened to an (∞, \cdot) assumption. Our proofs combine geometric and topological tools.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases convexity, (p, q) -theorem, k -transversal, infinite (p, q) -theorem

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.50

Funding Chaya Keller: Research partially supported by the Israel Science Foundation (grant no. 1065/20).

Acknowledgements The authors are grateful to Andreas Holmsen for valuable suggestions and information.

1 Introduction

1.1 Background

Helly's theorem and the (p, q) -theorem. The classical Helly's theorem [19] asserts that if \mathcal{F} is a family of compact convex sets in \mathbb{R}^d and every $d + 1$ (or fewer) members of \mathcal{F} have a non-empty intersection, then the whole family \mathcal{F} has a non-empty intersection.

For a pair of positive integers $p \geq q$, a family \mathcal{F} of sets in \mathbb{R}^d is said to satisfy the (p, q) -property if $|\mathcal{F}| \geq p$, none of the sets in \mathcal{F} is empty, and among every p sets of \mathcal{F} , some q have a non-empty intersection, or equivalently, can be pierced by a single point. A set $P \subset \mathbb{R}^d$ is called a *transversal* for \mathcal{F} if it has a non-empty intersection with every member of \mathcal{F} , or equivalently, if every member of \mathcal{F} is pierced by an element of P . In this language, Helly's theorem states that any family of compact convex sets in \mathbb{R}^d that satisfies the $(d + 1, d + 1)$ -property, has a singleton transversal.

One of the best-known generalizations of Helly's theorem is the (p, q) -theorem of Alon and Kleitman (1992), which resolved a 35-year old conjecture of Hadwiger and Debrunner [18].

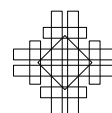
► **Theorem 1** (the (p, q) -theorem [3]). *For any triple of positive integers $p \geq q \geq d + 1$ there exists $c = c(p, q, d)$ such that if \mathcal{F} is a family of compact convex sets in \mathbb{R}^d that satisfies the (p, q) -property, then there exists a transversal for \mathcal{F} of size at most c .*



© Chaya Keller and Micha A. Perles;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 50; pp. 50:1–50:14
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



In the 30 years since the publication of the (p, q) -theorem, numerous variations, generalizations and applications of it were obtained (see, e.g., the surveys [13, 20]). We outline below three variations to which our results are closely related.

(p, q) -theorems for k -transversals. The question whether Helly's theorem can be generalized to k -transversals – namely, to piercing by k -dimensional flats (i.e., k -dimensional affine subspaces of \mathbb{R}^d) – goes back to Vincensini [32], and was studied extensively. Santaló [31] observed that there is no Helly-type theorem for general families of convex sets, even with respect to 1-transversals in the plane. Subsequently, numerous works showed that Helly-type theorems for 1-transversals and for $(d - 1)$ -transversals in \mathbb{R}^d can be obtained under additional assumptions on the sets of the family (see [20] and the references therein). A few of these results were generalized to k -transversals for all $1 \leq k \leq d - 1$ (see [5, 6]).

Concerning (p, q) -theorems, the situation is cardinaly different. In [1], Alon and Kalai obtained a (p, q) -theorem for *hyperplane transversals* (that is, for $(d - 1)$ -transversals in \mathbb{R}^d). The formulation of the theorem involves a natural generalization of the (p, q) -property:

For a family \mathcal{G} of objects (e.g., the family of all hyperplanes in \mathbb{R}^d), a family \mathcal{F} is said to satisfy the (p, q) -property with respect to \mathcal{G} if among every p members of \mathcal{F} , some q can be pierced by an element of \mathcal{G} . A set $P \subset \mathcal{G}$ is called a *transversal* for \mathcal{F} with respect to \mathcal{G} if every member of \mathcal{F} is pierced by an element of P .

► **Theorem 2** ([1]). *For any triple of positive integers $p \geq q \geq d + 1$ there exists $c = c(p, q, d)$ such that if \mathcal{F} is a family of compact convex sets in \mathbb{R}^d that satisfies the (p, q) -property with respect to piercing by hyperplanes, then there exists a hyperplane transversal for \mathcal{F} of size at most c .*

As an open problem at the end of their paper, Alon and Kalai [1] asked whether a similar result can be obtained for k -transversals, for $1 \leq k \leq d - 2$. The question was answered on the negative by Alon, Kalai, Matoušek and Meshulam [2], who showed by an explicit example that no such (p, q) -theorem exists for line transversals in \mathbb{R}^3 .

(p, q) -theorems without convexity. Numerous works obtained variants of the (p, q) -theorem in which the convexity assumption on the sets is replaced by a different (usually, topological) assumption. Most of the results in this direction base upon a result of Alon et al. [2], who showed that a (p, q) -theorem can be obtained whenever a *fractional Helly theorem* can be obtained, even without a convexity assumption on the elements of \mathcal{F} . In particular, the authors of [2] obtained a (p, q) -theorem for finite families of sets which are a *good cover*, meaning that the intersection of every sub-family is either empty or contractible. Matoušek [24] showed that bounded VC-dimension implies a (p, q) -theorem, and Pinchasi [29] proved a (p, q) -theorem for geometric hypergraphs whose ground set has a small union complexity. Recently, several more general (p, q) -theorems were obtained for families with a bounded Radon number, by Moran and Yehudayoff [26], Holmsen and Lee [21], and Patáková [28].

(p, q) -theorems for infinite set families. While most of the works on (p, q) -theorems concentrated on finite families of sets, several papers studied (p, q) -theorems for infinite set families.

It is well-known that Helly's theorem for infinite families holds under the weaker assumption that all sets are convex and closed, and at least one of them is bounded. In 1990, Erdős asked whether a (p, q) -theorem can be obtained in this weaker setting as well. Specifically, his conjecture – which was first published in [8] – was that a $(4, 3)$ -theorem holds for infinite families of convex closed sets in the plane in which at least one of the sets is bounded.

Following Erdős and Grünbaum, who refuted Erdős' conjecture and replaced it by a weaker conjecture of his own, several papers studied versions of the (p, q) -theorem for infinite families (see [25, 27]). These papers aimed at replacing the compactness assumption (which can be removed completely for finite families) by a weaker assumption.

1.2 Our contributions

In this paper we study variants of the (p, q) -theorem for infinite families \mathcal{F} of sets in \mathbb{R}^d . Our basic question is whether the assumption of the theorem can be replaced by the following weaker infinitary assumption, which we naturally call an (\aleph_0, q) -property: Among every \aleph_0 elements of \mathcal{F} , there exist some q that can be pierced by a single point (or more generally, by an element of \mathcal{G}). We show that despite the apparently weaker condition, (p, q) -theorems can be obtained in several settings of interest.

An $(\aleph_0, 2)$ -theorem for closed balls in \mathbb{R}^d . Our first result concerns the classical setting of point transversals and considers families of closed balls in \mathbb{R}^d . For such families, Danzer [9] obtained in 1956 a $(2, 2)$ -theorem in the plane, answering a question of Gallai. Grünbaum [17] obtained a $(2, 2)$ -theorem in \mathbb{R}^d , Kim et al. [22] obtained a $(p, 2)$ theorem in the plane for all $p \geq 2$, and finally, Dumitrescu and Jiang [11] obtained a $(p, 2)$ -theorem in \mathbb{R}^d for all $p \geq 2$. We show that an $(\aleph_0, 2)$ -theorem holds as well.

► **Theorem 3.** *Let \mathcal{F} be an infinite family of closed balls in \mathbb{R}^d . If among every \aleph_0 elements of \mathcal{F} , some two intersect, then \mathcal{F} can be pierced by a finite number of points.*

We note that unlike the standard (p, q) -theorems, there does not exist a universal constant $c = c(d)$ such that every family of closed balls in \mathbb{R}^d can be pierced by at most c points. Indeed, for any $m \in \mathbb{N}$, if the family consists of \aleph_0 copies of m pairwise disjoint balls then it satisfies the $(\aleph_0, 2)$ -property (and actually, even the much stronger (\aleph_0, \aleph_0) -property), yet it clearly cannot be pierced by less than m points.

An $(\aleph_0, k + 2)$ -theorem for “fat” sets in \mathbb{R}^d , with respect to k -transversals. Our main result concerns (p, q) -theorems with respect to k -transversals. In this setting, the construction presented in [2, Sec. 9] suggests that no $(\aleph_0, k + 2)$ -theorem with respect to k -transversals can be obtained for general families of convex sets in \mathbb{R}^d where $k < d - 1$, since even the stronger $(d + 1, d + 1)$ -property does not imply a bounded-sized k -transversal. However, we show that if the convexity assumption is replaced by an assumption that the elements of the family are “fat”,¹ then an $(\aleph_0, k + 2)$ -theorem can be obtained.

► **Definition 4.** *Let $0 < r \leq R$. A family \mathcal{F} of sets in \mathbb{R}^d is called (r, R) -fat if any $A \in \mathcal{F}$ contains a ball of radius r and is contained in a ball of radius R .*

► **Theorem 5.** *Let $0 < r \leq R$, $0 \leq k \leq d - 1$, and let \mathcal{F} be an infinite (r, R) -fat family of sets in \mathbb{R}^d . If among every \aleph_0 elements of \mathcal{F} , some $k + 2$ can be pierced by a k -flat, then \mathcal{F} can be pierced by a finite number of k -flats.*

¹ We note that a “fatness” assumption was considered in the context of (p, q) -theorems for families of convex sets in the plane, by Gao and Zerbib [16].

Theorem 5 allows significantly weakening the (p, q) -property assumption of “classical” (p, q) -theorems into an (∞, q) -property assumption, it applies to k -transversals for all $0 \leq k \leq d - 1$ (while the (p, q) -theorem for k -transversals holds only for $k = 0, d - 1$), and it does not require the sets in the family to be convex.

On the other hand, it requires a significant additional assumption – namely, that the elements of the family are “fat”. We show by an explicit construction that this assumption is essential.

► **Proposition 6.** *There exists an infinite family \mathcal{F} of open discs in the plane that satisfies the $(3, 3)$ -property (and so, also the $(\aleph_0, 3)$ -property) with respect to 1-transversals (i.e., piercing by lines), but cannot be pierced by a finite number of lines.*

Note that such a strong example could not be obtained for families of closed discs in the plane, since by Theorem 2, a family of compact convex sets in the plane that satisfies the $(3, 3)$ -property with respect to piercing by lines, admits a bounded-sized line transversal.

An infinite Ramsey-type theorem. In [23], Larman et al. observed that every $(p, 2)$ -theorem can be used to obtain a Ramsey-type theorem. Using a similar argument (presented in Sec. 7), Theorem 5 can be used to obtain the following Ramsey-type result.

► **Corollary 7.** *Let $0 < r \leq R$, $0 \leq k \leq d - 1$, and let \mathcal{F} be an infinite (r, R) -fat family of sets in \mathbb{R}^d . Denote $\alpha = |\mathcal{F}|$. Then one of the following holds:*

- *There exists $S \subset \mathcal{F}$ with $|S| = \aleph_0$ such that no $k + 2$ elements of S can be pierced by a k -flat.*
- *There exists $S' \subset \mathcal{F}$ with $|S'| = \alpha$, such that every $k + 2$ elements of S' can be pierced by a k -flat.*

For $\alpha > \aleph_0$ and $k \geq 1$, the assertion of Corollary 7 is significantly stronger than the best possible “generic” Ramsey theorem that can be obtained in the same setting. Indeed, the corresponding Ramsey-type theorem concerns (blue, red)-colorings of all r -element subsets of a set with cardinality α , for $r \geq 3$. In this setting, Erdős and Rado [14, Thm. 28] showed that in general, one cannot guarantee even the existence of either a set of $r + 1$ elements all of whose r -tuples are blue or a set of cardinality α all of whose r -tuples are red. Corollary 7 provides either an “all-blue” set with cardinality \aleph_0 or an “all-red” set with cardinality α (of course, for the specific coloring in which a $(k + 2)$ -tuple is colored blue if it can be pierced by a k -flat). This provides yet another example of the phenomenon that graphs and hypergraphs arising in geometry satisfy much stronger forms of Ramsey’s theorem than arbitrary graphs and hypergraphs. This phenomenon was demonstrated in several works in the finite setting (see [4, 7, 15, 23]), and our result provides an infinitary example.

Organization of the paper. In Section 2 we present some definitions, notations, and basic observations. In Section 3 we prove a lemma which shall be used in the proof of Theorem 5. Then, in Section 4 we prove Theorem 5. The construction of Proposition 6 is presented in Section 5, and the proof of Theorem 3 is given in Section 6. A more detailed comparison of Corollary 7 with generic Ramsey results is presented in Section 7. We conclude the paper with an open problem in Section 8.

2 Definitions, Notations, and Basic Observations

2.1 Definitions and notations

We use the following classical definitions.

- For $0 \leq k \leq d - 1$, a k -flat in \mathbb{R}^d is a k -dimensional affine subspace of \mathbb{R}^d (namely, a translation of a k -dimensional linear subspace of \mathbb{R}^d). In particular, a 0-flat is a point, a 1-flat is a line, and a $(d - 1)$ -flat is a hyperplane.
- The *direction* of a k -flat ($k > 0$) in \mathbb{R}^d is defined as follows. First, the k -flat is translated such that it will pass through the origin. Then, its direction is defined as the great $(k - 1)$ -sphere in which the k -flat intersects the sphere \mathcal{S}^{d-1} . (This definition follows [6].)
- A k_1 -flat and a k_2 -flat are called *parallel* if the direction of one of them is contained in the direction of the other. (Equivalently, this means that if both are translated so that they will pass through the origin, then one translation will be included in the other. Note that this relation is not transitive, and that two flats of the same dimension are parallel, if and only if one of them is a translation of the other.)
- For $\epsilon > 0$, an (open) ϵ -neighborhood of a point $x \in \mathcal{S}^{d-1}$ on the sphere is $B^\circ(x, \epsilon) \cap \mathcal{S}^{d-1}$, where $B^\circ(x, \epsilon)$ is the open ball with radius ϵ centered at x .
- A family $\mathcal{F} = \{B_\alpha\}_\alpha$ of sets in \mathbb{R}^d , is *independent w.r.t. k -flats* if no k -flat $\pi \subset \mathbb{R}^d$ intersects $k + 2$ B_α 's or more.

In the proofs of the theorems in the sequel, we mostly consider families \mathcal{F} of closed unit balls in \mathbb{R}^d , $d \geq 1$, no two of them are equal. We always assume w.l.o.g. that \mathcal{F} does not contain a ball centered at the origin, since all such balls are pierced by a single point, and hence by a single k -flat. We use the following definitions and notations:

- For $B = B(x, 1) \in \mathcal{F}$, the *direction* of B is the point $\hat{x} = x/\|x\|_2$. Of course, $\hat{x} \in \mathcal{S}^{d-1}$.
- For any $\hat{x} \in \mathcal{S}^{d-1}$ (which is not necessarily a direction of a ball in \mathcal{F}) and any $\epsilon > 0$, the (open) ϵ -neighborhood of \hat{x} in \mathcal{F} is

$$\mathcal{F}_{\hat{x}, \epsilon} = \{B(y, 1) \in \mathcal{F} : \hat{y} \in B^\circ(\hat{x}, \epsilon) \cap \mathcal{S}^{d-1}\},$$

that is, the set of all elements of \mathcal{F} whose directions are in an ϵ -neighborhood of \hat{x} .

- For convenience, we often focus on the point $\hat{x} = (0, 0, \dots, 0, 1) \in \mathcal{S}^{d-1}$, on the line $\ell = \{t\hat{x} : t \in \mathbb{R}\}$, and on projections onto the hyperplane orthogonal to ℓ (i.e., projections onto the first $d - 1$ coordinates.)

For each $B = B(x, 1) \in \mathcal{F}$, we denote by $B' \subset \mathbb{R}^{d-1}$ and $x' \in \mathbb{R}^{d-1}$ the projections of B and x , respectively. The d 'th coordinate of x , omitted in the projection, is denoted by $x(d)$.

2.2 Basic claims and observations

We use the two following simple claims.

▷ **Claim 8.** Let $\hat{x} = (0, 0, \dots, 0, 1)$, and let $\{B(x_n, 1)\}_{n=1,2,\dots}$ be a sequence of pairwise disjoint unit balls in \mathbb{R}^d such that $\lim_{n \rightarrow \infty} \hat{x}_n = \hat{x}$. Then $\lim_{n \rightarrow \infty} x_n(d) = \infty$.

Proof. Let $0 < M \in \mathbb{R}, \epsilon > 0$. There exists $n_1 \in \mathbb{N}$ such that for any $n > n_1$, \hat{x}_n is in the ϵ -neighborhood of \hat{x} . The set

$$\mathcal{F}_{\hat{x}, \epsilon} \cap \{B(x_n, 1) : n > n_1, x_n(d) < M\}$$

is contained in a finite area (which is a function of ϵ, d and M). By the disjointness of the balls in \mathcal{F} , there exists $n_2 > n_1$ such that for any $n > n_2$, $x_n(d) \geq M$. ◁

50:6 An $(\aleph_0, k + 2)$ -Theorem for k -Transversals

▷ **Claim 9.** Let $\mathcal{F} \subset \mathbb{R}^d$ be a family of balls of radius 1, and let G be a family of balls of radius $r > 0$, with the same centers. Then for any $0 \leq k \leq d - 1$, \mathcal{F} can be pierced by a finite set of k -flats if and only if G can be pierced by a finite set of k -flats.

Proof. Assume w.l.o.g. that $r > 1$. If \mathcal{F} can be pierced by finitely many k -flats, then the same clearly holds for G as well, as the elements of \mathcal{F} are contained in corresponding elements of G .

Assume that G can be pierced by finitely many k -flats, and take a finite family H of k -flats that pierces it. Replace each k -flat π in H by a sufficiently dense net of k -flats parallel to it, whose distance from π is at most $2r$. It is clear that the resulting finite family of k -flats pierces \mathcal{F} . ◀

3 A Technical Lemma

In the proof of Theorem 5, we shall need the following lemma.

► **Lemma 10.** Let \mathcal{F} be a family of closed unit balls in \mathbb{R}^d , let $0 \leq k \leq d - 1$, and let $\hat{x} = (0, 0, \dots, 0, 1)$. Assume that for any $\epsilon > 0$, the set $\mathcal{F}_{\hat{x}, \epsilon}$ cannot be pierced by a finite collection of k -flats.

Then there exists a sequence of balls, $\{B(x_n, 1)\}_{n=1,2,3,\dots} \subset \mathcal{F}$ such that $\lim_{n \rightarrow \infty} \hat{x}_n = \hat{x}$ and the sequence cannot be pierced by a finite family of k -flats.

We derive Lemma 10 from the following proposition.

► **Proposition 11.** Let \mathcal{F} be a family of closed unit balls in \mathbb{R}^d , let $0 \leq k \leq d - 1$ and $m \in \mathbb{N}$. If any finite subfamily of \mathcal{F} can be pierced by at most m k -flats, then \mathcal{F} can be pierced by at most m k -flats.

We first derive Lemma 10 from Proposition 11, and then present the proof of the proposition.

Proof of Lemma 10, assuming Proposition 11. Let \mathcal{F}, \hat{x} be as in the statement of the lemma, and assume that for any $\epsilon > 0$, the set $\mathcal{F}_{\hat{x}, \epsilon}$ cannot be pierced by a finite collection of k -flats.

We construct the sequence of balls $\{B(x_n, 1)\}_{n=1,2,3,\dots} \subset \mathcal{F}$ as follows. We take a sequence $\{\epsilon_m\}_{m=1,2,3,\dots}$, where $\epsilon_m = 1/m$. For each $m \in \mathbb{N}$, we find in $\mathcal{F}_{\hat{x}, \epsilon_m}$ a finite family G_m of balls that cannot be pierced by m k -flats (this is possible by Proposition 11). We define the sequence $\{B(x_n, 1)\}_{n=1,2,3,\dots}$ as $\bigcup_{m \in \mathbb{N}} G_m$. Namely, we arbitrarily order the balls in each G_m and add them to the sequence, allowing repetitions, starting with $m = 1$, proceeding to $m = 2$, etc.. We have $\lim_{n \rightarrow \infty} \hat{x}_n = \hat{x}$, since for any $\epsilon > 0$, only a finite number of $B(x_n, 1)$'s do not belong to $\mathcal{F}_{\hat{x}, \epsilon}$. Furthermore, $\{B(x_n, 1)\}_{n=1,2,3,\dots}$ cannot be pierced by m k -flats (for any $m \in \mathbb{N}$) since it contains the family G_m that cannot be pierced by m k -flats by its construction. Hence, $\{B(x_n, 1)\}_{n=1,2,3,\dots}$ cannot be pierced by a finite number of k -flats, as asserted. ◀

Proof of Proposition 11. Any k -flat $\pi \subset \mathbb{R}^d$ can be represented as

$$\pi = \{c + \lambda_1 v_1 + \dots + \lambda_k v_k : \lambda_i \in \mathbb{R}\},$$

where c is the point on π closest to the origin, and $\{v_1, \dots, v_k\}$ is an orthonormal basis of the vector subspace $\pi - \pi = \{x - y : x, y \in \pi\}$ which is parallel to π . (This actually means that the vector $\vec{0c}$ is orthogonal to each v_i .)

Assign to each k -flat π all the corresponding $(k + 1)$ -tuples of the type $\{c, v_1, \dots, v_k\}$. Note that while c is uniquely determined by π , the orthogonal basis is not. We obtain a representation of all k -flats in \mathbb{R}^d as $(k + 1)$ -tuples of d -vectors

$$\mathcal{A} = \{(c, v_1, \dots, v_k) : c, v_i \in \mathbb{R}^d \wedge \forall i \neq j, v_i \perp v_j \wedge v_i \perp c \wedge \|v_i\| = 1\} \subset \mathbb{R}^{d(k+1)}.$$

By the conditions of the proposition, we can assume w.l.o.g. that there exists a finite sub-family $\mathcal{F}_0 \subset \mathcal{F}$ that cannot be pierced by $m - 1$ k -flats. Let

$$\mathcal{A}^m = \{(c^1, v_1^1, \dots, v_k^1, \dots, c^m, v_1^m, \dots, v_k^m) : \forall 1 \leq j \leq m, (c^j, v_1^j, \dots, v_k^j) \in \mathcal{A}\} \subset \mathbb{R}^{d(k+1)m}$$

represent m -tuples of k -flats in \mathbb{R}^d .

Note that \mathcal{A}^m is not compact (as a subset of $\mathbb{R}^{d(k+1)m}$), since $\|c^j\|$ may be arbitrarily large. However, for any fixed closed unit ball $B = B(x_0, 1) \subset \mathbb{R}^d$, the subset $\Pi_B \subset \mathcal{A}^m$ that represents all m -tuples of k -flats intersecting $B \cup \mathcal{F}_0$, is a compact subset of \mathcal{A}^m . Indeed, all the m coordinates c^j , satisfy $\|c^j\| \leq \max\{\|x_0\| + 1, \max_{B' \in \mathcal{F}_0} \text{dist}(B', 0) + 2\}$.

Consider the family $\{\Pi_B\}_{B \in \mathcal{F}}$ where Π_B represents all m -tuples of k -flats that pierce $B \cup \mathcal{F}_0$. (For each such m -tuple of k -flats, we take all possible $(d(k + 1)m)$ -tuples that represent it.) Each Π_B is compact, and by the assumption, any finite sub-family $\{\Pi_{B_i}\}_{i=1}^n$ has non-empty intersection (that contains the representation of some m -tuple of k -flats that together intersect $\mathcal{F}_0, B_1, \dots, B_n$). Therefore, by the finite intersection property of compact sets, the whole family $\{\Pi_B\}_{B \in \mathcal{F}}$ has non-empty intersection. Any element in this non-empty intersection represents an m -tuple of k -flats that pierce together all the balls in \mathcal{F} . ◀

► Remark 12. Proposition 11 holds not only when \mathcal{F} is a family of unit balls, but actually for any family \mathcal{F} of non-empty compact sets in \mathbb{R}^d .

4 Proof of the Main Theorem

We restate Theorem 5, in a formulation that will be more convenient for the proof:

► **Theorem 5 (restated).** *Let $R, r > 0$ and let \mathcal{F} be a family of sets in \mathbb{R}^d such that each $S \in \mathcal{F}$ contains a ball of radius r and is contained in a ball of radius R . Let $0 \leq k \leq d - 1$. Then one of the two following conditions must hold:*

- \mathcal{F} can be pierced by a finite number of k -flats.
- \mathcal{F} contains an infinite sequence of sets that are independent w.r.t. k -flats (i.e., no k -flat pierces $k + 2$ of them).

First, we observe that it is sufficient to prove Theorem 5 for families of closed unit balls in \mathbb{R}^d .

▷ Claim 13 (Reduction to closed unit balls). If the assertion of Theorem 5 holds for all families of closed unit balls in \mathbb{R}^d , then it holds in the full generality stated in the theorem.

Proof. Let \mathcal{F} be a family as in the assumption. Construct a family \mathcal{F}_1 by taking, for each $S \in \mathcal{F}$, a closed ball of radius r contained in S . (Note that we can make sure that the centers of these balls are distinct, possibly at the price of reducing their radii to $r/2$.) Then, construct another family \mathcal{F}_2 by taking, for each $S \in \mathcal{F}$, a closed ball of radius $2R$ that contains S , with the same center as the corresponding ball in \mathcal{F}_1 .

Apply Theorem 5 to \mathcal{F}_2 . If it contains an infinite sequence of balls that are independent w.r.t. k -flats, then so does \mathcal{F} , since for each element of the sequence, we can take the element of \mathcal{F} that corresponds to it, and the resulting sequence of elements of \mathcal{F} will clearly be independent as well.

50:8 An $(\aleph_0, k + 2)$ -Theorem for k -Transversals

Otherwise, \mathcal{F}_2 can be pierced by a finite number of k -flats. Hence, by Claim 9, \mathcal{F}_1 can be pierced by a finite number of k -flats as well. This implies that \mathcal{F} can be pierced by a finite number of k -flats, since any element of \mathcal{F} contains an element of \mathcal{F}_1 . Therefore, the assertion of the theorem holds for \mathcal{F} . \triangleleft

By Claim 13, it is sufficient to prove Theorem 5 for families of closed unit balls in \mathbb{R}^d . A second reduction, before proceeding to the proof, is passing to *pairwise disjoint* unit balls.

\triangleright **Claim 14 (Reduction to pairwise disjoint balls).** If the assertion of Theorem 5 holds for all families of pairwise disjoint closed unit balls in \mathbb{R}^d , then it holds in the full generality stated in the theorem.

Proof. By Claim 13, it is sufficient to prove that if Theorem 5 holds for all families of pairwise disjoint closed unit balls in \mathbb{R}^d , then it holds for any family of closed unit balls.

Indeed, assume correctness for all families of pairwise disjoint closed unit balls in \mathbb{R}^d , and let \mathcal{F} be a family of arbitrary closed unit balls in \mathbb{R}^d . First, we pass to a subfamily $\bar{\mathcal{F}} \subset \mathcal{F}$ of pairwise disjoint balls, which is maximal under inclusion:

Consider the family \mathcal{G} of all subsets of \mathcal{F} in which all balls are pairwise disjoint. View \mathcal{G} as a poset with respect to inclusion. As each chain in \mathcal{G} has a maximal element (which is the union of its elements), by Zorn's lemma \mathcal{G} has a maximal element. This maximal element $\bar{\mathcal{F}} \subset \mathcal{F}$ is a set of pairwise disjoint balls, which is maximal under inclusion, among all the pairwise disjoint subfamilies.

By assuming correctness of Theorem 5 for families of pairwise disjoint closed unit balls, either $\bar{\mathcal{F}}$ contains an infinite sequence $\{\bar{\mathcal{F}}_n\}_{n \in \mathbb{N}}$ of balls that are independent w.r.t. k -flats, or $\bar{\mathcal{F}}$ can be pierced by a finite number of k -flats.

In the first case, $\{\bar{\mathcal{F}}_n\}_{n \in \mathbb{N}} \subset \bar{\mathcal{F}} \subset \mathcal{F}$ satisfies the second assertion of Theorem 5. In the second case, by the maximality of $\bar{\mathcal{F}}$, any ball in \mathcal{F} intersects some ball in $\bar{\mathcal{F}}$. Therefore, by replacing each k -flat in the finite piercing set of $\bar{\mathcal{F}}$, by a sufficiently dense net of k -flats surrounding it and parallel to it, we obtain a finite piercing set of k -flats for \mathcal{F} , that satisfies the first assertion of Theorem 5. \triangleleft

The proof of Theorem 5 is by induction, passing from $(k - 1, d - 1)$ to (k, d) . The induction basis is the case $k = 0$ of Theorem 5, reduced to a family of closed unit balls, by Claim 13. (The reduction to disjoint balls is not needed here.) We observe:

\blacktriangleright **Observation 15.** *Let \mathcal{F} be a family of (not necessarily disjoint) closed unit balls in \mathbb{R}^d . Then one of the two following conditions must hold:*

- \blacksquare \mathcal{F} can be pierced by a finite number of points.
- \blacksquare \mathcal{F} contains an infinite sequence of pairwise disjoint balls.

Proof. Consider the set $A = \{x \in \mathbb{R}^d \mid B(x, 1) \in \mathcal{F}\}$ of all centers of balls in \mathcal{F} . If A is bounded in some $B(0, R) \subset \mathbb{R}^d$, then clearly a finite set of points pierces all elements of \mathcal{F} . Otherwise, A is unbounded, hence \mathcal{F} contains an infinite sequence of pairwise disjoint balls, that can be obtained inductively. \blacktriangleleft

For $d = 1$, the assertion of Theorem 5, after applying the reduction of Claim 13, is exactly Observation 15. For $d \geq 2$, we shall prove the following version, which is sufficient due to the reductions of Claims 13 and 14:

► **Theorem 16.** *Let $d \geq 2$ and $0 \leq k \leq d-1$. Let \mathcal{F} be a family of pairwise disjoint closed unit balls in \mathbb{R}^d , and assume w.l.o.g. that \mathcal{F} does not contain a ball centered at the origin. Then:*

1. *If for any $\hat{x} \in \mathcal{S}^{d-1}$, there exists $\epsilon(\hat{x}) = \epsilon > 0$ such that $\mathcal{F}_{\hat{x}, \epsilon}$ can be pierced by finitely many of k -flats, then \mathcal{F} can be pierced by finitely many of k -flats.*
2. *If the condition of (1) does not hold, then \mathcal{F} contains an infinite sequence of balls that are independent w.r.t. k -flats (i.e., no k -flat pierces $k+2$ of them).*

Proof. First, we give the proof of the first assertion. Assume that for any $\hat{x} \in \mathcal{S}^{d-1}$, there exists $\epsilon = \epsilon(\hat{x}) > 0$ such that $\mathcal{F}_{\hat{x}, \epsilon}$ can be pierced by a finite number of k -flats. Pick such an $\epsilon(\hat{x})$ for each $\hat{x} \in \mathcal{S}^{d-1}$, and obtain an open covering of \mathcal{S}^{d-1} by open balls $B(\hat{x}, \epsilon(\hat{x}))$, for all $\hat{x} \in \mathcal{S}^{d-1}$.

By the compactness of the sphere, we can find a finite sub-cover, generated by balls around $\hat{x}_1, \dots, \hat{x}_n$. As each $\mathcal{F}_{\hat{x}_i, \epsilon(\hat{x}_i)}$ can be pierced by a finite number of k -flats, we can pierce all elements of \mathcal{F} by a finite collection of k -flats (which is the union of the k -flats that pierce $\mathcal{F}_{\hat{x}_i, \epsilon(\hat{x}_i)}$, for $i = 1, \dots, n$).

Now we move to the second assertion. Assume that for some $\hat{x} \in \mathcal{S}^{d-1}$ and for any $\epsilon > 0$, the family $\mathcal{F}_{\hat{x}, \epsilon}$ cannot be pierced by a finite number of k -flats. We assume w.l.o.g. that $\hat{x} = (0, 0, \dots, 1)$. We shall construct a sequence of elements of \mathcal{F} that is independent w.r.t. k -flats. The construction goes by induction, which reduces from k -flats in \mathbb{R}^d to $(k-1)$ -flats in \mathbb{R}^{d-1} .

Induction basis: $k = 0$. This case, which concerns piercing by points, follows by the argument of Observation 15.

Induction step: From $(k-1, d-1)$ to (k, d) . Assume that we proved the assertion for families in \mathbb{R}^{d-1} , with respect to piercing by $(k-1)$ -flats, and consider a family $\mathcal{F} \subset \mathbb{R}^d$ of pairwise disjoint closed unit balls.

First, we use Lemma 10 to find a sequence $G = \{B(x_n, 1)\}_{n=1,2,\dots}$ of elements of \mathcal{F} such that $\lim_{n \rightarrow \infty} \hat{x}_n = \hat{x}$ and the sequence cannot be pierced by a finite number of k -flats. From now on, we restrict ourselves to this sequence.

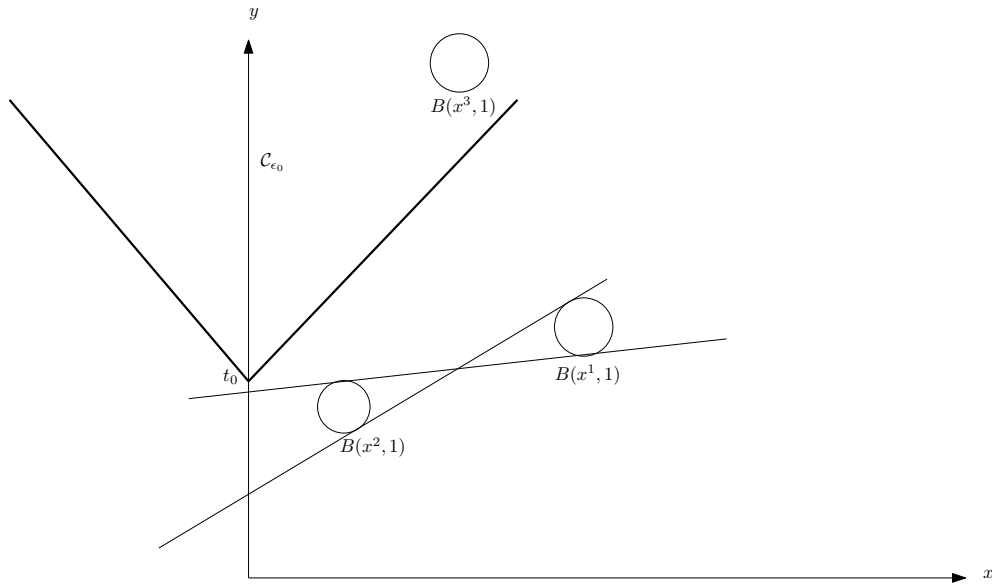
We project each $B(x_n, 1) \in G$ onto its first $d-1$ coordinates. Let the resulting set be G' , and similarly to the proof of Claim 14, let $G'' \subset G'$ be a subset of pairwise disjoint balls, maximal under inclusion in G' . By the induction hypothesis, either G'' (and therefore G') can be pierced by a finite number of $(k-1)$ -flats in \mathbb{R}^{d-1} , or else it contains a sequence of $(d-1)$ -dimensional balls that are independent w.r.t. $(k-1)$ -flats.

The first option cannot happen, as otherwise, one could pierce G with a finite number of k -flats (which are the pre-images of the $(k-1)$ -flats-transversal in \mathbb{R}^{d-1} under the projection), contrary to the choice of G . Hence, there exists a sub-sequence $\bar{G} = \{B(x_{n_l}, 1)\}_{l=1,2,\dots} \subset G$ of balls whose projections are independent w.r.t. $(k-1)$ -flats in \mathbb{R}^{d-1} . Note that as $\lim_{n \rightarrow \infty} \hat{x}_n = \hat{x}$, we have $\lim_{l \rightarrow \infty} \hat{x}_{n_l} = \hat{x}$. From now on, we restrict ourselves to this sequence and construct inductively a subsequence of it that will be independent w.r.t. k -flats in \mathbb{R}^d .

We construct the subsequence $\{B(x^n, 1)\}_{n=1}^\infty$ inductively. ($\{x^n\}_{n=1}^\infty$ is a subsequence of $\{x_{n_l}\}_{l=1}^\infty$.)

The first $k+1$ elements can be chosen arbitrarily. Assume that we already chose the balls $B(x^1, 1), \dots, B(x^m, 1)$, for $m \geq k+1$. To choose $B(x^{m+1}, 1)$, we first look at each $(k+1)$ -tuple of balls $(B(x^{i_1}, 1), \dots, B(x^{i_{k+1}}, 1))$ separately. By assumption, the corresponding projections on the first $d-1$ coordinates cannot be pierced by a $(k-1)$ -flat in \mathbb{R}^{d-1} . This implies that no k -flat that is parallel to the line $\ell = \{t\hat{x} : t \in \mathbb{R}\}$ can pierce all the $k+1$ balls $B(x^{i_1}, 1), \dots, B(x^{i_{k+1}}, 1)$.

50:10 An $(\aleph_0, k + 2)$ -Theorem for k -Transversals



■ **Figure 1** An illustration for the proof of Theorem 16 for $d = 2, k = 1$.

Consider the family U of all k -flats that pierce $(B(x^{i_1}, 1), \dots, B(x^{i_{k+1}}, 1))$. As none of them is parallel to ℓ , neither of their directions² contains the point $(0, 0, \dots, 0, 1) = \hat{x} \in \mathcal{S}^{d-1}$. By compactness of the elements of \mathcal{F} , this implies that there exists $\epsilon_0 > 0$, such that all these directions are disjoint with the ϵ_0 -neighborhood of \hat{x} on \mathcal{S}^{d-1} .

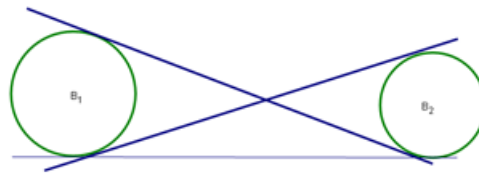
Now, let \mathcal{C}_{ϵ_0} be the unbounded cone whose vertex is the origin and whose intersection with \mathcal{S}^{d-1} is the boundary of $\epsilon_0/2$ -neighborhood of \hat{x} on \mathcal{S}^{d-1} . (Informally, this is a cone of small aperture around the positive direction of the d 'th axis.) We claim that there exists $t_0 \in \mathbb{R}$ such that for any $t > t_0$, the translation $(0, 0, \dots, 0, t) + \mathcal{C}_{\epsilon_0}$ is disjoint from all k -flats in U (see Figure 1).

To see this, for each k -flat $L \in U$ we define a function $f_L : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ by $f_L(t) = \min\{\|x'\| : x \in L \wedge x(d) = t\}$ (for all the relevant notations, see the end of Section 2.1). It is clear that f_L attains a minimum, and that since L is not parallel to ℓ , this minimum is attained in a single point, $t = \operatorname{argmin}(f_L) \in \mathbb{R}$. Now, we define a function $g : U \rightarrow \mathbb{R}$ by $g(L) = \operatorname{argmin}(f_L)$. By compactness of the elements of \mathcal{F} , this function attains a maximum, t_0 . As the direction of any $L \in U$ is disjoint with the ϵ_0 -neighborhood of \hat{x} on \mathcal{S}^{d-1} , it follows that $L \cap ((0, 0, \dots, 0, t) + \mathcal{C}_{\epsilon_0}) = \emptyset$, for all $t > t_0$.

We are now ready to choose the ball $B(x^{m+1}, 1)$. We go over all $(k + 1)$ -tuples of balls $(B(x^{i_1}, 1), \dots, B(x^{i_{k+1}}, 1))$ with $1 \leq i_1 < i_2 < \dots < i_{k+1} \leq m$. For each of them, we find (ϵ_0, t_0) such that for any $t > t_0$, any k -flat that pierces $(B(x^{i_1}, 1), \dots, B(x^{i_{k+1}}, 1))$ is disjoint with the cone $(0, 0, \dots, 0, t) + \mathcal{C}_{\epsilon_0}$, where \mathcal{C}_{ϵ_0} is as defined above.

Let ϵ_1 be the minimum of the ϵ_0 values, and let t_1 be the maximum of the t_0 values. It is clear that if we make sure that $B(x^{m+1}, 1)$ is entirely included in the cone $(0, 0, \dots, 0, t_1 + 1) + \mathcal{C}_{\epsilon_1}$, then no k -flat will pierce both $B(x^{m+1}, 1)$ and a $(k + 1)$ -tuple $(B(x^{i_1}, 1), \dots, B(x^{i_{k+1}}, 1))$. We can indeed choose $B(x^{m+1}, 1)$ in this way, by Claim 8. This completes the proof. ◀

² See Section 2.1 for the needed definitions.



■ **Figure 2** An illustration for Section 5.

5 Proof of Proposition 6

In this section we prove Proposition 6. Namely, we construct an infinite family of open discs in the plane that satisfies the (3, 3)-property with respect to line transversals, but cannot be pierced by a finite number of lines.

Proof of Proposition 6. Let $\mathcal{F} = \{\mathcal{F}_n\}_{n=1}^\infty \subset \mathbb{R}^2$, where $\mathcal{F}_n = B(n, 1/n)$ is an open disc centered at $(n, \frac{1}{n})$ with radius $\frac{1}{n}$. The family \mathcal{F} does not admit a finite line transversal, since the x -axis meets no element of \mathcal{F} , any line that is parallel to the x -axis meets finitely many elements of \mathcal{F} , and any line that forms a positive angle with the x -axis, intersects a finite subfamily of \mathcal{F} .

On the other hand, any $\mathcal{F}' \subset \mathcal{F}$ which is independent w.r.t. lines, satisfies $|\mathcal{F}'| \leq 2$. Indeed, consider the two leftmost discs $B_1, B_2 \in \mathcal{F}'$. The right wedge that the two common inner tangents of B_1 and B_2 form, contains all elements of \mathcal{F} that are to the right of B_1 and B_2 (see Figure 2). Therefore, any element of \mathcal{F} that lies to the right of B_1 and B_2 is pierced by a line that passes through B_1 and B_2 , and hence cannot be contained in \mathcal{F}' . ◀

We note that no similar example could be constructed with closed balls, since by the Danzer-Grünbaum-Klee theorem [10], such a family would be pierced by a single line.

6 An 0-flat Transversal With no Restriction on the Radii

In this section we prove Theorem 3, which is a much stronger version of Observation 15. This stronger version holds with no restriction on the radii. Let us restate the theorem in a formulation which is more convenient for us:

► **Theorem 3 (restated).** *Let \mathcal{F} be a family of closed balls in \mathbb{R}^d (with no restriction on the radii). Then one of the two following conditions must hold:*

- \mathcal{F} can be pierced by a finitely many points.
- \mathcal{F} contains an infinite sequence of pairwise disjoint balls.

Before proceeding into the proof, we prove a reduction to the case where all elements of \mathcal{F} are contained in a closed bounded ball $B(0, R) \subset \mathbb{R}^d$.

▷ **Claim 17.** Let $R > 0$ and assume we proved Theorem 3 where any ball in \mathcal{F} is contained in $B(0, R)$. Then Theorem 3 holds.

Proof. Define the distance of a closed ball $B \subset \mathbb{R}^d$ from the origin, $dist(B, 0)$, as the Euclidian distance between the origin and the point $x \in B$ which is closest to the origin.

If the set $\{dist(B, 0) : B \in \mathcal{F}\}$ is unbounded in \mathbb{R}^d , then one can inductively construct an infinite sequence of pairwise disjoint balls in \mathcal{F} , whose distance from the origin tends to infinity.

50:12 An $(\aleph_0, k + 2)$ -Theorem for k -Transversals

From now on we assume that there exists some $0 < R \in \mathbb{R}$ such that for any $B \in \mathcal{F}$, $\text{dist}(B, 0) \leq R - 2$. Replace each $B \in \mathcal{F}$ whose radius $r(B) > 1$, by some closed smaller ball $B' \subset B$ with $r(B') = 1$, such that $\text{dist}(0, B) = \text{dist}(0, B')$. Let \mathcal{F}' be the obtained family. Any ball in \mathcal{F}' is contained in $B(0, R)$.

By the assumption of our claim, either \mathcal{F}' can be pierced by finitely many points, or \mathcal{F}' contains an infinite sequence $\mathcal{F}'' \subset \mathcal{F}'$ of pairwise disjoint balls. In the first case, the finite piercing set of \mathcal{F}' pierces \mathcal{F} as well.

In the second case, remove from \mathcal{F}'' all balls with radius 1. There are only finitely many such balls, since $\mathcal{F}'' \subset \mathcal{F}' \subset B(0, R)$, and the elements of \mathcal{F}'' are pairwise disjoint. After removing from \mathcal{F}'' all balls with radius 1, we are left with an infinite subfamily of balls each of which belongs to \mathcal{F} (since the transition from \mathcal{F} to \mathcal{F}' involved only the radius-1 balls of \mathcal{F}'), which are pairwise disjoint. \triangleleft

Proof of Theorem 3. By Claim 17 we can assume that there exists $R > 0$ such that each ball in \mathcal{F} is contained in $B(0, R)$. We can assume w.l.o.g. that \mathcal{F} contains no ball of radius 0. Indeed, if \mathcal{F} contains finitely many such balls, we can remove them without changing the assertion. Otherwise, \mathcal{F} contains an infinite sequence of radius-0 balls, and then we are done again.

Each $x \in B(0, R)$ is of exactly one of the two following types:

Type (a): For each $\delta > 0$, there exists some $B \in \mathcal{F}$, $B \cap B^\circ(x, \delta) \neq \emptyset$, with $r(B) < \delta$ and $B \cap \{x\} = \emptyset$.

Type (b): There exists $0 < \delta = \delta(x)$ such that for any $B \in \mathcal{F}$ with $B \cap B^\circ(x, \delta) \neq \emptyset$, the following holds: Either $r(B) \geq \delta$ or $B \cap \{x\} \neq \emptyset$.

If $B(0, R)$ contains some point x of type (a), then there exists an infinite sequence of pairwise disjoint balls in \mathcal{F} (that tends to $\{x\}$). Indeed, start with $\delta_0 = 1$ and pick some $B_0 \in \mathcal{F}$, $B_0 \cap B^\circ(x, \delta_0) \neq \emptyset$, with $r(B_0) < \delta_0$ and $B_0 \cap \{x\} = \emptyset$. Since B_0 is closed, it has a positive distance ϵ from x . Let $\delta_1 = \frac{\epsilon}{10}$ and pick some $B_1 \in \mathcal{F}$, $B_1 \cap B^\circ(x, \delta_1) \neq \emptyset$, $r(B_1) < \delta_1$ and $B_1 \cap \{x\} = \emptyset$. Continue in the same manner to construct an infinite sequence of pairwise disjoint balls in \mathcal{F} .

The remaining case is where each $x \in B(0, R)$ is of type (b). Then for each $x \in B(0, R)$ there exists $0 < \delta = \delta(x)$ such that any $B \in \mathcal{F}$ that intersects $B^\circ(x, \delta)$ can be pierced by finitely many points, say, by $f(x)$ points. (Note that the exact value of $f(x)$ depends on the choice of $\delta = \delta(x)$.) By the finite intersection property of compact sets in \mathbb{R}^d , the open cover $\bigcup_{x \in B(0, R)} B^\circ(x, \delta(x))$ of $B(0, R)$ has a finite sub-cover $B(0, R) \subset \bigcup_{i=1}^k B^\circ(x_i, \delta(x_i))$. Since all the balls in \mathcal{F} that intersect $B^\circ(x_i, \delta(x_i))$ can be pierced by $f(x_i)$ points, it follows that all the elements of \mathcal{F} can be pierced by at most $\sum_{i=1}^k f(x_i)$ points. \triangleleft

7 Comparison of Corollary 7 with Generic Infinite Ramsey-type Theorems

We begin with a restatement of Corollary 7.

► **Corollary 7 (restated).** Let $0 < r \leq R$, $0 \leq k \leq d - 1$, and let \mathcal{F} be an infinite (r, R) -fat family of sets in \mathbb{R}^d . Denote $\alpha = |\mathcal{F}|$. Then one of the following holds:

- There exists $S \subset \mathcal{F}$ with $|S| = \aleph_0$ s.t. no $k + 2$ elements of S can be pierced by a k -flat.
- There exists $S' \subset \mathcal{F}$ with $|S'| = \alpha$, s.t. every $k + 2$ elements of S' can be pierced by a k -flat.

Proof. If the first condition does not hold, then \mathcal{F} satisfies the $(\aleph_0, k + 2)$ property, and hence by Theorem 5, \mathcal{F} can be pierced by a finite number of k -flats L_1, L_2, \dots, L_n . Denote $\mathcal{F}_i = \{A \in \mathcal{F} : A \cap L_i \neq \emptyset\}$. At least one of the families \mathcal{F}_i is of cardinality α , and every $k + 2$ elements of it can be pierced by a k -flat. \triangleleft

For $\alpha = \aleph_0$, Corollary 7 is not interesting, as it follows directly from the infinite Ramsey theorem [30]. For $\alpha > \aleph_0$ and $k = 0$ (i.e., piercing by points), Corollary 7 is already significantly stronger than the conclusion of the “diagonal” Ramsey’s theorem, which guarantees only a countable monochromatic subset. However, it is still uninteresting since it follows from the Erdős-Dushnik-Miller theorem [12], which asserts that for any infinite α , any (blue, red)-coloring of a graph on α vertices contains either a monochromatic blue set of cardinality \aleph_0 or a monochromatic red set of cardinality α .

The interesting case is $k \geq 1$ – i.e., piercing by k -flats with $k \geq 1$, which is the hard case in Theorem 5. Here, the corresponding Ramsey-type theorem concerns (blue, red)-colorings of all r -element subsets of a set with cardinality α , for $r \geq 3$. In this setting, Erdős and Rado [14, Thm. 28] showed that in general, one cannot guarantee even the existence of either a set of $r + 1$ elements all of whose r -tuples are blue or a set of cardinality α all of whose r -tuples are red. Corollary 7 provides either an “all-blue” set with cardinality \aleph_0 or an “all-red” set with cardinality α (of course, for the specific coloring in which a $(k + 2)$ -tuple is colored blue if it can be pierced by a k -flat).

Therefore, in its “main” setting of $k \geq 1$, Theorem 5 provides an infinite Ramsey theorem which is significantly stronger than the best possible “generic” Ramsey theorems. Moreover, the assertion of Corollary 7 cannot be strengthened to obtain a first possibility with $|S| > \aleph_0$, since once no $k + 2$ elements of S can be pierced by a k -flat, all elements of S must be pairwise disjoint; hence $|S| \leq \aleph_0$.

8 Open Problem

A natural open problem which arises in light of Theorem 3 and Proposition 6 is, whether an $(\aleph_0, k + 2)$ -theorem (like Theorem 5) can be obtained for families of closed balls, without the “fatness” assumption. For $1 \leq k < d - 1$, such a theorem cannot be obtained for general families of compact convex sets, as shown by the construction of Alon et al. [2]. However, it still might hold for families of balls.

References

- 1 N. Alon and G. Kalai. Bounding the piercing number. *Discrete Comput. Geom.*, 13:245–256, 1995.
- 2 N. Alon, G. Kalai, J. Matoušek, and R. Meshulam. Transversal numbers for hypergraphs arising in geometry. *Adv. Appl. Math.*, 29:79–101, 2002.
- 3 N. Alon and D. J. Kleitman. Piercing convex sets and the Hadwiger-Debrunner (p,q) -problem. *Advances in Mathematics*, 96(1):103–112, 1992.
- 4 N. Alon, J. Pach, R. Pinchasi, R. Radoičić, and M. Sharir. Crossing patterns of semi-algebraic sets. *J. Combin. Theory, Ser. A*, 111:310–326, 2005.
- 5 J. L. Arocha, J. Bracho, and L. Montejano. Flat transversals to flats and convex sets of a fixed dimension. *Advances in Mathematics*, 213(2):902–918, 2007.
- 6 B. Aronov, J. E. Goodman, and R. Pollack. A Helly-type theorem for higher-dimensional transversals. *Comput. Geom.*, 21:177–183, 2002.
- 7 I. Bárány and G. Kalai. Helly-type problems, 2021. [arXiv:2108.08804](https://arxiv.org/abs/2108.08804).
- 8 V. Boltyanski and A. Soifer. *Geometric études in combinatorial mathematics*. Center for Excellence in Mathematical Education, Colorado Springs, CO, 1991.
- 9 L. Danzer. Zur lösung des gallaischen problems über kreisscheiben in der euklidischen ebene. *Studia Sci. Math. Hungar.*, 21:111–134, 1986.
- 10 L. Danzer, B. Grünbaum, and V. Klee. Helly’s theorem and its relatives. In V. Klee, editor, *Convexity, Proceedings of Symposium in Pure Mathematics*, volume 7, pages 100–181. American Mathematical Society, Providence, RI, 1963.

- 11 A. Dumitrescu and M. Jiang. Piercing translates and homothets of a convex body. *Algorithmica*, 61(1):94–115, 2011.
- 12 B. Dushnik and E. W. Miller. Partially ordered sets. *American Journal of Mathematics*, 63(3):600–610, 1941.
- 13 J. Eckhoff. A survey of the Hadwiger-Debrunner (p, q) -problem. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry*, volume 25 of *Algorithms and Combinatorics*, pages 347–377. Springer Berlin Heidelberg, 2003.
- 14 P. Erdős and R. Rado. A partition calculus in set theory. *Bulletin of the American Mathematical Society*, 62(5):427–489, 1956.
- 15 J. Fox, J. Pach, and C. D. Tóth. Intersection patterns of curves. *J. London Math. Society*, 83:389–406, 2011.
- 16 S. Gao and S. Zerbib. The $(2,2)$ and $(4,3)$ properties in families of fat sets in the plane. *SIAM Journal of Discrete Math.*, 33(3):1326–1337, 2019.
- 17 B. Grünbaum. On intersections of similar sets. *Portugaliae Mathematica*, 18:155–164, 1959.
- 18 H. Hadwiger and H. Debrunner. Über eine variante zum Hellyschen satz. *Archiv der Mathematik*, 8(4):309–313, 1957.
- 19 E. Helly. Über mengen konvexer körper mit gemeinschaftlichen punkte. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 32:175–176, 1923.
- 20 A. Holmsen and R. Wenger. Helly-type theorems and geometric transversals. In J. O’Rourke, J. E. Goodman and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry, 3rd Edition*, pages 91–123. CRC Press LLC, Boca Raton, FL, 2017.
- 21 A. F. Holmsen and D. Lee. Radon numbers and the fractional Helly theorem. *Isr. J. Math.*, 241:433–447, 2021.
- 22 S. J. Kim, K. Nakprasit, M.J. Pelsmajer, and J. Skokan. Transversal numbers of translates of a convex body. *Discrete Math.*, 306:2166–2173, 2006.
- 23 D. Larman, J. Matoušek, J. Pach, and J. Töröcsik. A ramsey-type result for planar convex sets. *Bulletin of London Math. Soc.*, 26:132–136, 1994.
- 24 J. Matoušek. Bounded VC-dimension implies a fractional Helly theorem. *Discrete Comput. Geom.*, 31(2):251–255, 2004.
- 25 A. Montejano, L. Montejano, E. Roldán-Pensado, and P. Soberón. About an Erdős–Grünbaum conjecture concerning piercing of non-bounded convex sets. *Discrete Comput. Geom.*, 53(4):941–950, 2015.
- 26 S. Moran and A. Yehudayoff. On weak ϵ -nets and the Radon number. *Discret. Comput. Geom.*, 64(4):1125–1140, 2020.
- 27 T. Müller. A counterexample to a conjecture of Grünbaum on piercing convex sets in the plane. *Discrete Math.*, 313(24):2868–2871, 2013.
- 28 Z. Patáková. Bounding Radon number via Betti numbers. In *36th International Symposium on Computational Geometry, SoCG 2020*, volume 164 of *LIPICs*, pages 61:1–61:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- 29 R. Pinchasi. A note on smaller fractional Helly numbers. *Discrete Comput. Geom.*, 54:663–668, 2015.
- 30 F. P. Ramsey. On a problem of formal logic. *Proc. London Math. Soc. (Ser. 2)*, 30(1):264–286, 1930.
- 31 L. Santaló. Un teorema sobre conjuntos de paralelepipedos de aristas paralelas. *Publ. Inst. Mat. Univ. Nac. Litoral*, 2:49–60, 1940.
- 32 P. Vincensini. Figures convexes et variétés linéaires de l’espace euclidien à n dimensions. *Bull Sci. Math.*, 59:163–174, 1935.

Farthest-Point Voronoi Diagrams in the Presence of Rectangular Obstacles

Mincheol Kim ✉


Department of Computer Science and Engineering,
Pohang University of Science and Technology, South Korea

Chanyang Seo ✉

Graduate School of Artificial Intelligence,
Pohang University of Science and Technology, South Korea

Taehoon Ahn ✉

Department of Computer Science and Engineering,
Pohang University of Science and Technology, South Korea

Hee-Kap Ahn ✉ 

Graduate School of Artificial Intelligence, Department of Computer Science and Engineering,
Pohang University of Science and Technology, South Korea

Abstract

We present an algorithm to compute the geodesic L_1 farthest-point Voronoi diagram of m point sites in the presence of n rectangular obstacles in the plane. It takes $O(nm + n \log n + m \log m)$ construction time using $O(nm)$ space. This is the first optimal algorithm for constructing the farthest-point Voronoi diagram in the presence of obstacles. We can construct a data structure in the same construction time and space that answers a farthest-neighbor query in $O(\log(n + m))$ time.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Geodesic distance, L_1 metric, farthest-point Voronoi diagram

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.51

Related Version *Full Version*: <http://arxiv.org/abs/2203.03198>

Funding This research was partly supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00905, Software Star Lab (Optimal Data Structure and Algorithmic Applications in Dynamic Geometric Environment)) and (No. 2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH)).

1 Introduction

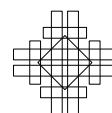
A Voronoi diagram of a set of sites is a subdivision of the space under consideration into subspaces by assigning points to sites with respect to a certain proximity. Typical Voronoi assignment models are the nearest-point model and the farthest-point model where every point is assigned to its nearest site and its farthest site, respectively. There are results for computing Voronoi diagrams in the plane [1, 13, 14, 24], under different metrics [9, 17, 18, 23], or for various types of sites [2, 8, 22].

For m point sites in the plane, the nearest-point and farthest-point Voronoi diagrams of the sites can be constructed in $O(m \log m)$ time [14, 24]. When the sites are contained in a simple polygon with no holes, the distance between any two points in the polygon, called the *geodesic distance*, is measured as the length of the shortest path contained in the polygon and connecting the points (called the *geodesic path*). There has been a fair amount of work computing the geodesic nearest-point and farthest-point Voronoi diagrams of m point sites



© Mincheol Kim, Chanyang Seo, Taehoon Ahn, and Hee-Kap Ahn;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 51; pp. 51:1–51:15
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



in a simple n -gon [3, 4, 20, 21] to achieve the lower bound $\Omega(n + m \log m)$ [3]. Recently, optimal algorithms of $O(n + m \log m)$ time were given for the geodesic nearest-point Voronoi diagram [19] and for the geodesic farthest-point Voronoi diagram [25].

The problem of computing Voronoi diagrams is more challenging in the presence of obstacles. Each obstacle plays as a hole and there can be two or more geodesic paths connecting two points avoiding those holes. The geodesic nearest-point Voronoi diagram of m point sites can be computed in $O(m \log m + k \log k)$ time by applying the continuous Dijkstra paradigm [16], where k is the number of total vertices of obstacles. However, no optimal algorithm is known for the farthest-point Voronoi diagram in the presence of obstacles in the plane, even when the obstacles are of elementary shapes such as axis-aligned line segments and rectangles. The best result of the geodesic farthest-point Voronoi diagram known so far takes $O(mk \log^2(m + k) \log k)$ time by Bae and Chwa [5]. They also showed that the total complexity of the geodesic farthest-point Voronoi diagram is $\Theta(mk)$.

In the presence of n rectangular obstacles under L_1 metric, there are some work for farthest-neighbor queries. Ben-Moshe et al. [7] presented a data structure with $O(nm \log(n + m))$ construction time and $O(nm)$ space for m point sites that supports farthest point queries in $O(\log(n + m))$ time. They also showed that the L_1 geodesic farthest-point Voronoi diagram has complexity $\Theta(nm)$, but without presenting any algorithm for computing the diagram. Later Ben-Moshe et al. [6] gave a tradeoff between the query time and the preprocessing/space such that a data structure of size $O((n+m)^{1.5})$ can be constructed in $O((n+m)^{1.5} \log^2(n+m))$ to support farthest point queries in $O((n+m)^{0.5} \log(n+m))$ time.

The geodesic center of a set of objects in a polygonal domain is the set of points in the domain that minimize the maximum geodesic distance from input objects. Thus, it can be obtained once the geodesic farthest-point Voronoi diagram of the objects is constructed. For m points in the presence of n axis-aligned rectangular obstacles in the plane, Choi et al. [10] showed that the geodesic center of the points under the L_1 metric consists of $\Theta(nm)$ connected regions and they gave an $O(n^2m)$ -time algorithm to compute the geodesic center. Later, Ben-Moshe et al. [7] gave an $O(nm \log(n + m))$ -time algorithm for the problem.

Our Result. In this paper, we present an algorithm that computes the geodesic L_1 farthest-point Voronoi diagram of m points in the presence of n rectangular obstacles in the plane in $O(nm + n \log n + m \log m)$ time using $O(nm)$ space. The running time and space complexity of our algorithm match the time and space bounds of the Voronoi diagram. Thus, it is the first optimal algorithm for computing the geodesic farthest-point Voronoi diagram in the presence of obstacles.

To do this, we construct a data structure for L_1 farthest-neighbor queries in $O(nm + n \log n + m \log m)$ time using $O(nm)$ space. This improves upon the results by Ben-Moshe et al. [7], and the construction time and space are the best among the data structures supporting $O(\log(n+m))$ query time for L_1 farthest neighbors. Then we present an optimal algorithm to compute the explicit geodesic L_1 farthest-point Voronoi diagram in $O(nm + n \log n + m \log m)$ time using $O(nm)$ space, which matches the time and space lower bounds of the diagram.

As a byproduct, we compute the geodesic center under the L_1 metric in $O(nm + n \log n + m \log m)$ time. This result improves upon the algorithm by Ben-Moshe et al. [7].

Outline. First, we construct four farthest-point maps, one for each of the four axis directions, either the x - or y -axis, and either positive or negative. In the course, we construct a data structure for L_1 farthest-neighbor queries in $O(nm + n \log n + m \log m)$ time using $O(nm)$ space. For each axis direction, we apply the plane sweep technique with a line orthogonal to

the direction and moving along the direction. During the sweep, we maintain the status of the sweep line in a balanced binary search tree and its associated structures while handling events induced by the point sites and the sides of rectangles parallel to the sweep line. There are m events induced by point sites and $O(n)$ events induced by rectangles. After sorting the events in $O(n \log n + m \log m)$ time, we show that we can handle all events induced by point sites in $O(nm)$ time. Additionally, we show that each event induced by a rectangle can be handled in $O(m + \log n)$ time. By the plane sweep, we construct a data structure consisting of $O(n + m)$ line segments parallel to the sweep line and $O(nm)$ points in $O(nm + n \log n + m \log m)$ time in total. Given a query, it uses axis-aligned ray shooting queries on the data structure to find the farthest site from the query. The four farthest-point maps are planar subdivisions, and they can be constructed during the plane sweep in the same time and space.

With the four farthest-point maps and the data structure for farthest-neighbor queries, we construct the geodesic L_1 farthest-point Voronoi diagram explicitly. First, we decompose the plane, excluding the holes, into rectangular faces using vertical line segments, each extended from a vertical side of a hole. Then, we partition each face in the decomposition into zones such that the farthest-point Voronoi diagram restricted to a zone coincides with the corresponding region of a farthest-point map. This partition is done by using the boundary between two farthest-point maps, which can be computed by traversing the cells in the two maps in which the boundary lies. Finally, we glue the corresponding regions along the boundaries of zones, and then glue all adjacent faces along their boundaries to obtain the geodesic L_1 farthest-point Voronoi diagram. We show that this can be done in $O(nm + n \log n + m \log m)$ time in total.

For the centers of m points in the presence of n axis-aligned rectangles in the plane, we can find them from the farthest-point Voronoi diagram in time linear to the complexity of the diagram.

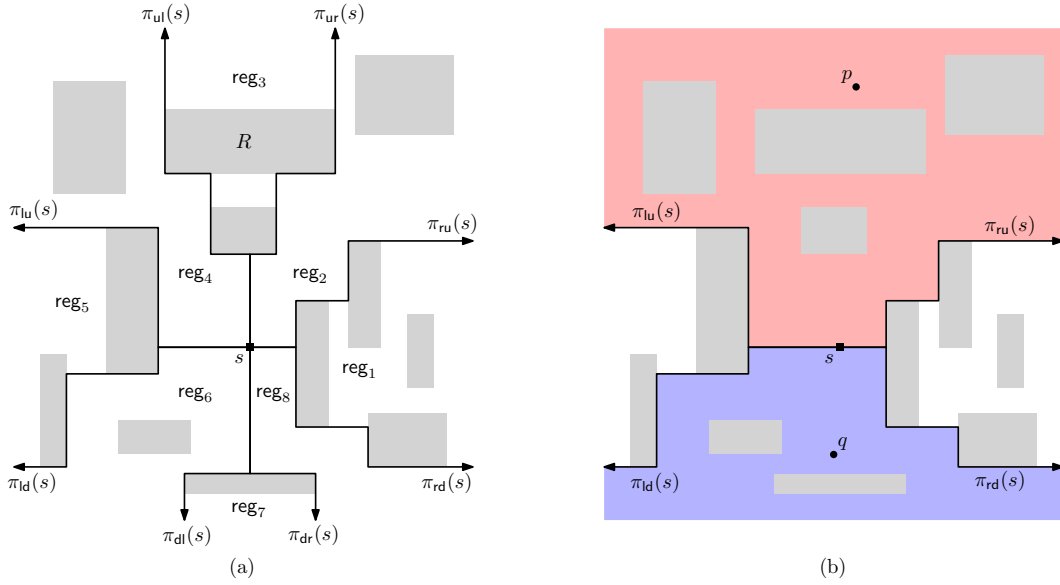
2 Preliminaries

Let R be a set of n open disjoint rectangles and S be a set of m point sites lying in the *free space* $F = \mathbb{R}^2 - \bigcup_{R \in R} R$. We consider the L_1 metric. For ease of description, we omit L_1 . We use $x(p)$ and $y(p)$ to denote the x -coordinate and y -coordinate of a point p , respectively. For two points p and q in F , we use pq to denote the line segment connecting them. Whenever we say a path connecting two points in F , it is a path contained in F . There can be more than one geodesic path connecting two points p and q avoiding the holes. We use $\pi(p, q)$ to denote a fixed geodesic path connecting p and q , and use $d(p, q)$ to denote the geodesic distance between p and q , which is the length of $\pi(p, q)$.

We make a general position assumption that no point in F is equidistant from four or more distinct sites. We use $f(p)$ to denote the set of sites of S that are farthest from a point $p \in F$ under the geodesic distance, that is, a site s is in $f(p)$ if and only if $d(s, p) \geq d(s', p)$ for all $s' \in S$. If there is only one farthest site, we use $f(p)$ to denote the site.

A horizontal line segment ℓ can be represented by the two x -coordinates $x_1(\ell)$ and $x_2(\ell)$ of its endpoints ($x_1(\ell) < x_2(\ell)$) and the y -coordinate $y(\ell)$ of them. For an axis-aligned rectangle R , let $x_1(R)$ and $x_2(R)$ denote the x -coordinates of the left and right sides of R .

A path is *x -monotone* if and only if the intersection of the path with any line perpendicular to the x -axis is connected. Likewise, a path is *y -monotone* if and only if the intersection of the path with any line perpendicular to the y -axis is connected. A path is *xy -monotone* if and only if the path is x -monotone and y -monotone. Observe that if a path connecting two points is xy -monotone, it is a geodesic path connecting the points.



■ **Figure 1** Gray rectangles are holes. (a) The eight paths partition F into eight regions $\text{reg}_1, \dots, \text{reg}_8$. Region reg_3 consists of two regions separated by a rectangle R . (b) Every geodesic path from s to p is y^+ -monotone and p is y^+ -reachable from s . Every geodesic path from s to q is y^- -monotone and q is y^- -reachable from s .

2.1 Eight Monotone Paths from a Point

Choi and Yap [11] gave a way of partitioning the plane with rectangular holes into eight regions using eight xy -monotone paths from a point. We use their method to partition F as follows. Consider a horizontal ray emanating from a point $s = p_1 \in F$ going rightwards. The ray stops when it hits a rectangle $R \in \mathcal{R}$ at a point p'_1 . Let p_2 be the top-left corner of R . We repeat this process by taking a horizontal ray from p_2 going rightwards until it hits a rectangle, and so on. Then we obtain an xy -monotone path $\pi_{ru}(s) = p_1 p'_1 p_2 p'_2 \dots$ from s that alternates going *rightwards* and going *upwards*.

By choosing two directions, one going either rightwards or leftwards horizontally, and one going either upwards or downwards vertically, and ordering the chosen directions, we define eight rectilinear xy -monotone paths with directions: rightwards-upwards (ru), upwards-rightwards (ur), upwards-leftwards (ul), leftwards-upwards (lu), leftwards-downwards (ld), downwards-leftwards (dl), downwards-rightwards (dr), and rightwards-downwards (rd). Let $\pi_\delta(s)$ denote one of the eight paths corresponding to the direction δ in $\{ru, ur, ul, lu, ld, dl, dr, rd\}$.

Some of the eight paths $\pi_\delta(s)$ may overlap in the beginning from s but they do not cross each other. The paths partition F into eight regions $\text{reg}_1, \dots, \text{reg}_8$ with the indices sorted around s in a counterclockwise order such that reg_1 denotes the region lying to the right of s , below $\pi_{ru}(s)$ and above $\pi_{rd}(s)$. Observe that reg_2 is not necessarily connected. See Figure 1(a) for an illustration.

► **Lemma 1** ([11, 12]). *Every geodesic path connecting two points is either x -, y -, or xy -monotone. For a point $s \in F$, following three statements hold.*

- *If $p \in \text{reg}_1 \cup \text{reg}_5$, every geodesic path from s to p is x -monotone but not y -monotone.*
- *If $p \in \text{reg}_3 \cup \text{reg}_7$, every geodesic path from s to p is y -monotone but not x -monotone.*
- *If $p \in \text{reg}_2 \cup \text{reg}_4 \cup \text{reg}_6 \cup \text{reg}_8 \cup \Pi(s)$, every geodesic path from s to p is xy -monotone, where $\Pi(s)$ is the union of the eight paths $\pi_\delta(s)$.*

Based on Lemma 1, we define a few more terms. For any point p in $\text{reg}_2 \cup \text{reg}_3 \cup \text{reg}_4$ (and the boundaries of the regions), we say p is y^+ -reachable from s , and every geodesic path from s to p is y^+ -monotone. Any point $q \in \text{reg}_6 \cup \text{reg}_7 \cup \text{reg}_8$ (and the boundaries of the regions) is y^- -reachable from s , and every geodesic path from s to q is y^- -monotone. See Figure 1(b). Similarly, any point $p \in \text{reg}_1 \cup \text{reg}_2 \cup \text{reg}_8$ (and the boundaries of the regions) is x^+ -reachable from s , and every geodesic path from s to p is x^+ -monotone. Any point $q \in \text{reg}_4 \cup \text{reg}_5 \cup \text{reg}_6$ (and the boundaries of the regions) is x^- -reachable from s , and every geodesic path from s to q is x^- -monotone.

3 Farthest-point Maps

Based on Lemma 1 and the four directions of monotone paths in the previous section, we define four *farthest-point maps*. A farthest-point map $M_{y^+} = M_{y^+}(S)$ of S in F corresponding to the positive y -direction is a planar subdivision of F into cells. For a point $p \in F$, a site $s \in S$ is a farthest site of p in M_{y^+} if $d(p, s) \geq d(p, s')$ for every site $s' \in S$ from which p is y^+ -reachable. If p is y^+ -reachable from no site in S , p has no farthest site in M_{y^+} . Thus, a cell of M_{y^+} is defined on $F \setminus C_\emptyset$, where C_\emptyset denotes the set of points of F that are y^+ -reachable from no site in S . A site s corresponds to one or more cells in M_{y^+} with the property that a point $p \in F \setminus C_\emptyset$ lies in a cell of s if and only if $d(p, s) > d(p, s')$ for every $s' \in S \setminus \{s\}$ from which p is y^+ -reachable.

We define M_{y^-} , M_{x^+} and M_{x^-} analogously with respect to their corresponding directions. Since the four maps have the same structural and combinatorial properties with respect to their corresponding directions, we describe only M_{y^+} in the following. Let B be an axis-aligned rectangular box such that S , R , and all vertices of the four farthest-point maps are contained in the interior of B . We focus on $F \cap B$ only, and use F as $F \cap B$.

In the following, we analyze the edges of M_{y^+} using the bisectors of pairs of sites. Let $F(s, s')$ denote a set of points of F that are y^+ -reachable from two sites s and s' . To be specific, $F(s, s')$ is an intersection of two regions, one lying above $\pi_{lu}(s)$ and $\pi_{ru}(s)$ and the other lying above $\pi_{lu}(s')$ and $\pi_{ru}(s')$. Thus, the boundary of $F(s, s')$ coincides with the upper envelope of $\pi_{lu}(s)$, $\pi_{ru}(s)$, $\pi_{lu}(s')$ and $\pi_{ru}(s')$. We use $F(s, s)$ to denote the set of points that are y^+ -reachable from a site s .

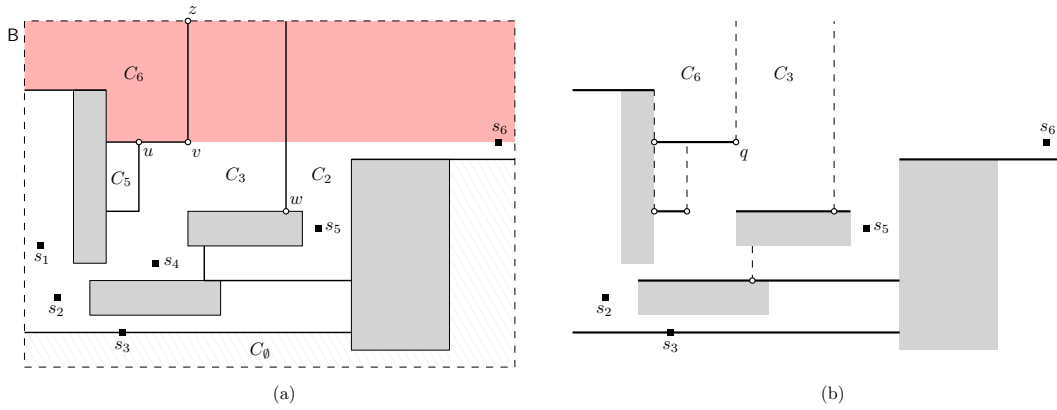
For any two distinct sites $s, s' \in S$, their *bisector* consists of all points $x \in F$ satisfying $\{x \mid d(x, s) = d(x, s')\}$. Observe that the bisector may contain a two-dimensional region. We use $b(s, s')$ to denote the line segments and the boundary of the two-dimensional region in the bisector of s and s' .

A proof of the following lemma is given in the full version.

► **Lemma 2.** *For any two sites s and s' , $b(s, s') \cap F(s, s')$ consists of axis-aligned segments.*

Let $f_\delta(p)$ denote the set of farthest sites from a point $p \in F$ among the sites from which p is δ -reachable for $\delta \in \{y^+, y^-, x^+, x^-\}$. For each horizontal segment of $\pi_{lu}(s) \cup \pi_{ru}(s)$, we call the portion h of the segment such that $f_{y^+}(p) = \{s\}$ for any point $p \in h$, a *b-edge*. Observe that no point p' with $x_1(h) \leq x(p') \leq x_2(h)$ and $y(p') = y(h) - \varepsilon$ for any $\varepsilon > 0$ is y^+ -reachable from s . Thus, a *b-edge* is also an edge of M_{y^+} . Since every edge of M_{y^+} is part of a bisector of two sites in S or a *b-edge*, it is either horizontal or vertical. See Figure 2(a).

► **Corollary 3.** *Every edge of M_{y^+} is an axis-aligned line segment.*



■ **Figure 2** (a) M_{y^+} for $S = \{s_1, \dots, s_6\}$ restricted to a box B with four rectangular holes (gray). s_i has a corresponding cell C_i for $i = 2, 3, 5, 6$ while s_1 and s_4 have no cell. A vertical edge uz is from $b(s_3, s_6)$ in the (red) region $F(s_3, s_6)$. A horizontal edge uv is not part of $b(s_3, s_6)$ but it is part of a b -edge as no point lying below uv is y^+ -reachable from s_6 . (b) Illustration of Q_{y^+} corresponding to M_{y^+} . At the boundary point q , $d(q, s_3) = d(q, s_6)$.

For sites contained in a simple polygon, Aronov et al. [4] gave a lemma, called *Ordering Lemma*, that the order of sites along their convex hull is the same as the order of their Voronoi cells along the boundary of a simple polygon. We give a lemma on the order of sites in the presence of rectangular obstacles. We use it in analyzing the maps and Voronoi diagrams. A proof of the following lemma is given in the full version.

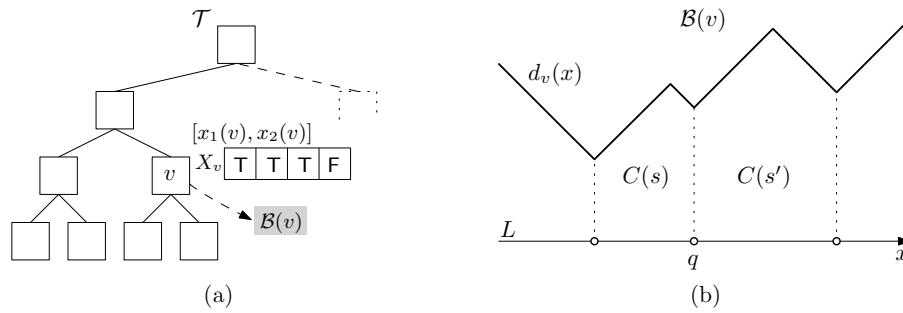
► **Lemma 4.** *Let pq be a horizontal segment contained in $F \setminus C_\emptyset$ with $x(p) < x(q)$. For any two sites $f_p \in f(p)$ and $f_q \in f(q)$ such that p and q are y^+ -reachable from both f_p and f_q , if $f_p \notin f(q)$ or $f_q \notin f(p)$, $x(f_p) > x(f_q)$.*

Since there are at most m sites, we obtain the following corollary from Lemma 4.

► **Corollary 5.** *Any horizontal line segment contained in F intersects at most m cells in M_{y^+} .*

Using Corollary 3 and 5, we analyze the complexity of M_{y^+} as follows. Note that each lower endpoint of a vertical edge of M_{y^+} appears on a horizontal line segment passing through a site or the top side of a rectangle. By Corollary 5, the maximal horizontal segment through the top side of a rectangle in R and contained in F intersects $O(m)$ vertical edges of M_{y^+} . Moreover, the maximal horizontal line segment through a site s and contained in F intersects $O(1)$ lower endpoints of vertical edges on the boundary of the cell of s . Since there are n rectangles in R and m sites in S , M_{y^+} has $O(nm + m) = O(nm)$ vertical edges. Every horizontal edge of M_{y^+} is a segment of a bisector or a b -edge, and it is incident to a side of a rectangle or another vertical edge. Since there are $O(n)$ rectangle sides, and $O(1)$ horizontal edges of M_{y^+} that are incident to a vertical edge, M_{y^+} has $O(n + nm) = O(nm)$ horizontal edges. Thus, M_{y^+} has complexity $O(nm)$.

Now we show that every farthest site $s \in f(p)$ of a point p in F is one of the farthest sites of p in the four farthest-point maps. By the definition of the farthest-point maps, p is contained in a cell of M_{y^+} , M_{y^-} , M_{x^+} or M_{x^-} . Since every geodesic path connecting two points is either y^+ -, y^- -, x^+ -, or x^- -monotone by Lemma 1, $s \in f(p)$ is one of the farthest sites of p in the four farthest-point maps. If p is contained in cells of two or more maps, we compare their distances to the farthest sites defining the cells and take the ones with the largest distance as the farthest sites of p . Thus, once the four farthest-point maps are constructed, the farthest sites of a query point can be computed from the map.



■ **Figure 3** (a) Illustration of a balanced binary search tree \mathcal{T} . A node v in \mathcal{T} has domain $[x_1(v), x_2(v)]$, array X_v , and a pointer to $\mathcal{B}(v)$. (b) Illustration of $\mathcal{B}(v)$ and $d_v(x)$.

4 Data Structure for Farthest-neighbor Queries

We present an algorithm that constructs a data structure for farthest site queries. We denote m point sites of S by s_1, \dots, s_m such that $x(s_1) \leq \dots \leq x(s_m)$, and n rectangular obstacles of R by R_1, \dots, R_n . The data structure consists of four parts, each for one axis direction. Since the four parts can be constructed in the same way with respect to their directions, we focus on the part corresponding to the positive y -direction, and thus the structure corresponds to M_{y^+} . We use Q_{y^+} to denote the query data structure.

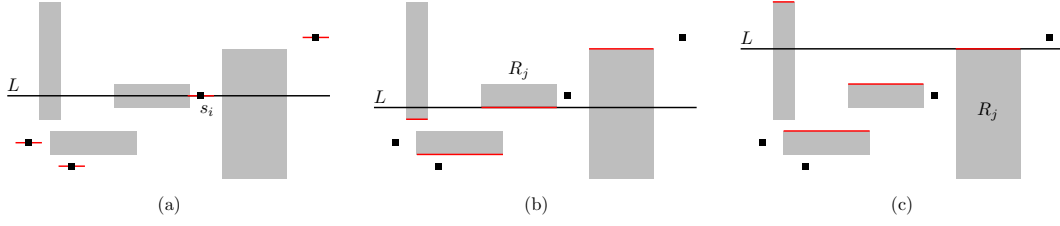
By Corollary 3, we can find the farthest site of a query point using a vertical ray shooting query to the horizontal edges of M_{y^+} and a binary search on the lower endpoints of vertical edges of M_{y^+} lying on the horizontal edges of M_{y^+} . Thus, we construct Q_{y^+} such that it consists of the horizontal edges of M_{y^+} and the endpoints of vertical edges of M_{y^+} lying on the horizontal edges of M_{y^+} .

A point q lying on a horizontal segment h of Q_{y^+} is the lower endpoint of a vertical edge of M_{y^+} if and only if there are two points $q_1 = (x(q) - \varepsilon, y(q))$ and $q_2 = (x(q) + \varepsilon, y(q))$ for sufficiently small $\varepsilon > 0$ satisfying $f_{y^+}(q_1) \cup f_{y^+}(q_2) = f_{y^+}(q)$ and $f_{y^+}(q_1) \neq f_{y^+}(q_2)$. We call each lower endpoint of vertical edges lying on h a *boundary point* on h . See Figure 2(b).

We use a plane sweep algorithm with a horizontal sweep line L to construct the horizontal line segments in Q_{y^+} . Note that $F \cap L$ consists of disjoint horizontal segments along L . The status of L is the sequence of segments in $F \cap L$ along L . The status changes while L moves upwards over the plane, but not continuously. Each update of the status occurs at a particular y -coordinate, which we call an *event*. To do such updates efficiently, we maintain three data structures for L : a *balanced binary search tree* \mathcal{T} representing the status, a *boundary list* \mathcal{B} , and a list \mathcal{D} of distance functions. The structures \mathcal{B} and \mathcal{D} are associated structures of \mathcal{T} .

We store the segments of $F \cap L$ in a balanced binary search tree \mathcal{T} in increasing order of x -coordinate of their left endpoints. Each node v of \mathcal{T} corresponds to a horizontal line segment h_v of $F \cap L$. We store $x_1(h_v)$ and $x_2(h_v)$, and an array X_v of m Boolean variables at v . We set $X_v[i] = \text{T}$ if a point on h_v is y^+ -reachable from s_i for $i = 1 \dots, m$. Otherwise, we set $X_v[i] = \text{F}$. The range of v is $[x_1(v), x_2(v)]$ for $x_1(v) = x_1(h_v)$ and $x_2(v) = x_2(h_v)$. There are at most $n + 1$ nodes in \mathcal{T} , and each node maintains an array of size $O(m)$, so \mathcal{T} itself uses $O(nm)$ space in total. See Figure 3(a).

The list \mathcal{B} consists of boundary lists $\mathcal{B}(v)$ for nodes v of \mathcal{T} . Each node v of \mathcal{T} has a pointer to its boundary list $\mathcal{B}(v)$, which is a doubly-linked list of the boundary points (including the endpoints of h_v) lying on h_v . Each boundary point in \mathcal{B} is the intersection of L and a vertical edge of M_{y^+} , so there are $O(nm)$ boundary points in \mathcal{B} .



■ **Figure 4** Three types of events. (a) site events. (b) bottom-side events. (c) top-side events.

Let $d_\delta(p) = d(s, p)$ for a site $s \in f_\delta(p)$ if $f_\delta(p) \neq \emptyset$, or $d_\delta(p) = -\infty$ for $\delta \in \{y^+, y^-, x^+, x^-\}$. The list \mathcal{D} consists of distance functions d_v for nodes v of \mathcal{T} . Let $p(r)$ denote a point on L with $x(p(r)) = r$ for a real number r . Each node v of \mathcal{T} has a pointer to its distance function $d_v(x) = d_{y^+}(p(x))$ for x in the range $[x_1(v), x_2(v)]$ of v . It is a piecewise linear function with pieces (segments) of slopes 1 or -1 . See Figure 3(b).

There are three types of events: (1) a site event, (2) a bottom-side event, and (3) a top-side event. A site event occurs when L encounters a site in \mathcal{S} . A bottom-side event occurs when L encounters the bottom side of a rectangle in \mathcal{R} . A top-side event occurs when L encounters the top side of a rectangle in \mathcal{R} . Thus, there are m site events, n bottom-side events, and n top-side events. See Figure 4.

We maintain and update \mathcal{T} , \mathcal{B} and \mathcal{D} during the plane sweep for those events. To handle events, we first sort the events in y -coordinate order, which takes $O((n+m)\log(n+m)) = O(n\log n + m\log m)$ time. We update $d_v(x)$ only at those events and keep it unchanged between two consecutive events. To reflect the distances from sites to $p(x) \in h_v$ correctly, we assign an additive weight to $d_v(x)$, which is the difference in the y -coordinates between the current event and the last event at which $d_v(x)$ is updated.

Initially, when L is at the bottom side of \mathcal{B} , \mathcal{T} consists of one node v with $x_1(v) = x_1(\mathcal{B})$, $x_2(v) = x_2(\mathcal{B})$, and $X_v[i] = \mathbf{F}$ for all $i \in \{1, \dots, m\}$. $\mathcal{B}(v)$ has no boundary point and $d_v(x) = -\infty$ for all x , since no points on L is y^+ -reachable from any sites.

4.1 Handling a site event

When L encounters a site $s_i \in \mathcal{S}$, we find the node $v \in \mathcal{T}$ such that $x_1(v) \leq x(s_i) \leq x_2(v)$. Every point on h_v is y^+ -reachable from s_i , so we set $X_v[i] = \mathbf{T}$. We can find v in $O(\log n)$ time, and set $X_v[i] = \mathbf{T}$ in constant time. Thus, it takes $O(\log n)$ time to update \mathcal{T} .

For any point $p(x) \in h_v$, $d(s_i, p(x)) = |x - x(s_i)|$. By Lemma 4, there is at most one maximal interval $I \subset [x_1(v), x_2(v)]$ such that $d_v(x) < d(s_i, p(x))$ for every $x \in I$. Moreover, I is bounded from left by $x_1(v)$ or from right by $x_2(v)$ because $d_v(x)$ is continuous and consists of pieces (segments) of slopes 1 or -1 , and $d(s_i, p(x)) = |x - x(s_i)|$. We find the boundary point $p(x^*) \in h_v$ induced by s_i such that $d_v(x^*) = d(s_i, p(x^*))$. If I is bounded from left, we update $d_v(x)$ to $d_v(x) = d(s_i, p(x))$ for $x \leq x^*$. If I is bounded from right, we update $d_v(x)$ to $d_v(x) = d(s_i, p(x))$ for $x \geq x^*$.

If there is no such point $p(x^*)$, either $d_v(x) < d(s_i, p(x))$ or $d_v(x) > d(s_i, p(x))$ for all x with $x_1(v) \leq x \leq x_2(v)$. If $d_v(x) < d(s_i, p(x))$, we update $d_v(x)$ to $d_v(x) = d(s_i, p(x))$ for $x_1(v) \leq x \leq x_2(v)$. If $d_v(x) > d(s_i, p(x))$, we do not update $d_v(x)$.

We update $\mathcal{B}(v)$ by removing all the boundary points of $\mathcal{B}(v)$ lying in the interior of I in time linear to the number of the boundary points, and then inserting $p(x^*)$ into $\mathcal{B}(v)$.

Since there are m site events, it takes $O(m \log n)$ time in total to update \mathcal{T} . The total time to remove the boundary points is linear to the total number of boundary points in \mathcal{Q}_{y^+} , which is $O(nm)$.

► **Lemma 6.** *We can handle all site events in $O(nm)$ time using $O(nm)$ space.*

4.2 Handling a bottom-side event

When L encounters the bottom side of a rectangle $R \in \mathcal{R}$, the line segment of $F \cap L$ incident to the bottom side is replaced by two line segments by the event. See Figure 4(b). Thus, we update \mathcal{T} by finding the node $v \in \mathcal{T}$ with $x_1(v) \leq x_1(R) < x_2(R) \leq x_2(v)$, removing v from \mathcal{T} , and then inserting two new nodes u and w into \mathcal{T} . We set $(x_1(u), x_2(u)) = (x_1(v), x_1(R))$, $(x_1(w), x_2(w)) = (x_2(R), x_2(v))$, $X_u = X_v$, and $X_w = X_v$. This takes $O(\log n)$ time since \mathcal{T} is a balanced binary search tree. It takes $O(m)$ time to copy the Boolean values of X_v to X_u and X_w , and to remove X_v . Thus, it takes $O(m + \log n)$ time to update \mathcal{T} .

We update \mathcal{B} by inserting two lists $\mathcal{B}(u)$ and $\mathcal{B}(w)$ into \mathcal{B} , copying the boundary points of $\mathcal{B}(v)$ to the lists, and then removing $\mathcal{B}(v)$ from \mathcal{B} . By Corollary 5, h_v intersects $O(m)$ cells in \mathcal{M}_{y^+} . Thus, $\mathcal{B}(v)$ has $O(m)$ boundary points, and the update to $\mathcal{B}(u)$ and $\mathcal{B}(w)$ takes $O(m)$ time. There is no change to distance functions.

Since there are n bottom-side events, it takes $O(nm + n \log n)$ time to update \mathcal{T} and $O(nm)$ time to update \mathcal{B} for all bottom-side events.

► **Lemma 7.** *We can handle all bottom-side events in $O(nm + n \log n)$ time using $O(nm)$ space.*

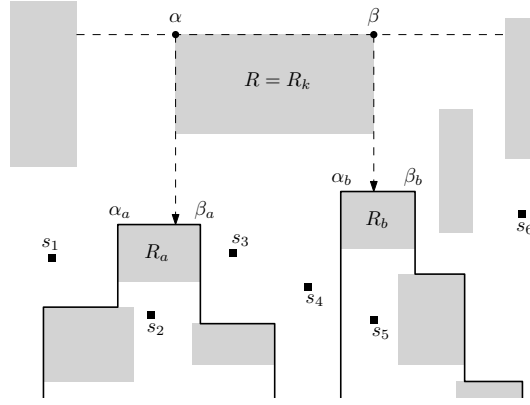
4.3 Handling a top-side event

When L encounters the top side of a rectangle $R \in \mathcal{R}$, the two consecutive segments in $F \cap L$ incident to R are replaced by one segment spanning them by the event. See Figure 4(c). We update \mathcal{T} by finding the two nodes $u, w \in \mathcal{T}$ with $x_2(u) = x_1(R)$ and $x_1(w) = x_2(R)$, removing u and w from \mathcal{T} , and then inserting a new node v into \mathcal{T} . We set $x_1(v) = x_1(u)$, $x_2(v) = x_2(w)$, and $X_v[i] = X_u[i] \vee X_w[i]$ for each $i = 1, \dots, m$. This takes $O(m + \log n)$ time.

We update the distance function $d_v(x)$ for x with $x_1(v) \leq x \leq x_1(R)$ as follows. The geodesic path from any point $p(x) \in h_u$ to s_i with $X_u[i] = \mathbf{F}$ and $X_w[i] = \mathbf{T}$ is xy -monotone by Lemma 1, and thus $d(s_i, p(x)) = y(p(x)) - y(s_i) + |x(s_i) - x|$. Also, we observe that $x(s_i) \geq x$ for any x . Thus, every $p(x)$ has the same site s^* as its farthest site among the sites s_i with $X_u[i] = \mathbf{F}$ and $X_w[i] = \mathbf{T}$. Then $d(s^*, p(x)) = y(p(x)) - y(s^*) + x(s^*) - x$. By Lemma 4, there is at most one maximal interval I of $x \in [x_1(v), x_1(R)]$ such that $d_v(x) \leq d(s^*, p(x))$. Moreover, I is bounded from left by $x_1(v)$. We find the boundary point $p(x^*) \in h_u$ such that $d_v(x^*) = d(s^*, p(x^*))$, and update $d_v(x)$ to $d(s^*, p(x))$ for $x \leq x^*$.

If there is no such point $p(x^*)$, either $d_v(x) < d(s^*, p(x))$ or $d_v(x) > d(s^*, p(x))$ for all x with $x_1(v) \leq x \leq x_1(R)$. If $d_v(x) < d(s^*, p(x))$, we update $d_v(x)$ to $d_v(x) = d(s^*, p(x))$ for $x_1(v) \leq x \leq x_1(R)$. If $d_v(x) > d(s^*, p(x))$, we do not update $d_v(x)$.

We update $\mathcal{B}[x_1(v), x_1(R)]$, which is a part of $\mathcal{B}(v)$ with range $[x_1(v), x_1(R)]$, by removing all the boundary points in the interior of I in time linear to the number of the boundary points, and then inserting $p(x^*)$ as a boundary point. We can handle the case of x with $x_2(R) \leq x \leq x_2(v)$, and update $\mathcal{B}[x_2(R), x_2(v)]$ analogously.



■ **Figure 5** $S^T = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ is partitioned into $S_k = \{s_2, s_3, s_4, s_5\}$, $S(\alpha) = \{s_1\}$, and $S(\beta) = \{s_6\}$. For two rectangles R_a and R_b , $S_a = \{s_2\}$ and $S_b = \{s_5\}$.

Computing distance functions for a top side

We show how to compute $d_v(x)$ for $x \in [x_1(R), x_2(R)]$ and update $\mathcal{B}[x_1(R), x_2(R)]$ efficiently.

For an index k , let α_k and β_k denote the top-left corner and the top-right corner of $R_k \in \mathcal{R}$, and let S_k denote the set of the sites that lie below the polygonal curve consisting of $\pi_{dl}(\alpha_k)$, the top side of R_k , and $\pi_{dr}(\beta_k)$.

For the top-side event of $R = R_k$, let $\alpha = \alpha_k$ and $\beta = \beta_k$. Note that $x(\alpha) = x_1(R)$ and $x(\beta) = x_2(R)$. Let S^T be the set of the sites s_i , with $X_v[i] = \text{T}$ for all $i = 1, \dots, m$. We partition S^T into three disjoint subsets, S_k , $S(\alpha)$, and $S(\beta)$, such that $S(\alpha) = \{s_i \in S^T \setminus S_k \mid x(s_i) \leq x_1(R)\}$ and $S(\beta) = \{s_i \in S^T \setminus S_k \mid x(s_i) \geq x_2(R)\}$. See Figure 5.

Every geodesic path from any site in $S(\alpha)$ or $S(\beta)$ to any point on the top side of R is xy -monotone. Thus for any point $p(x)$ lying on the top side of R , we can compute $d(s^\alpha, p(x))$ and $d(s^\beta, p(x))$, where s^α and s^β are the farthest sites of $p(x)$ among sites in $S(\alpha)$ and among sites in $S(\beta)$, respectively, as we did for $\mathcal{B}[x_1(v), x_1(R)]$ or $\mathcal{B}[x_2(R), x_2(v)]$.

We denote by $d_\alpha(i, x) = d(\alpha, s_i) + x - x(\alpha)$ the length of a geodesic path from a site s_i to $p(x)$ passing through α , and denote by $d_\beta(i, x) = d(\beta, s_i) + x(\beta) - x$ the length of a geodesic path from s_i to $p(x)$ passing through β . Let $D(x) = \max_{s_i \in S_k} \min\{d_\alpha(i, x), d_\beta(i, x)\}$ for all x with $x(\alpha) \leq x \leq x(\beta)$. Then $d_v(x) = \max\{d(s^\alpha, p(x)), D(x), d(s^\beta, p(x))\}$. Thus, once we compute $D(x)$ in $O(m)$ time, we can compute $d_v(x)$ in time linear to the complexity of $D(x)$, which is $O(m)$. To compute $D(x)$, we find the two rectangles hit first by the vertical rays, one emanating from α and one emanating from β , going downwards. Using these two rectangles we compute the distance functions $d(\alpha, s_i)$ and $d(\beta, s_i)$ for all $s_i \in S_k$. Using these distance functions, we can compute $D(x)$ in $O(m)$ time. Details are given in the full version. We update $\mathcal{B}[x_1(R), x_2(R)]$ in $O(m)$ time using $d_v(x)$.

There are n top-side events, so we can handle the top-side events in $O(nm + n \log n)$ time. In addition, we compute distances from $O(m)$ sites to each corner of $O(n)$ rectangles, and store them. Using ray shooting queries emanating from the corners of rectangles, it takes $O(nm) + O(n \log n)$ time using $O(nm)$ space. Therefore, we have the following lemma.

► **Lemma 8.** *We can handle all top-side events in $O(nm + n \log n)$ time using $O(nm)$ space.*

4.4 Constructing the query data structure

Initially, $Q_{y^+} = \emptyset$. For each site event and top-side event, we update $d_v(x)$ and $\mathcal{B}(v)$ for node v of \mathcal{T} corresponding to the event. We insert a horizontal segment h corresponding to each interval which is updated at the event into Q_{y^+} , and copy the boundary points into h . For each site event, at most one horizontal line segment h is inserted. There is no boundary point in the interior of h , so we can copy h with two endpoints in $O(1)$ time. For each top-side event, at most three horizontal line segments are inserted. They have $O(m)$ boundary points by Lemma 4, so we can copy them in $O(m)$ time. There are $O(n + m)$ horizontal segments and $O(nm)$ boundary points in Q_{y^+} , so the query structure Q_{y^+} uses $O(nm)$ space.

Farthest-point queries

Once Q_{y^+} is constructed, we can find $f_{y^+}(q)$ from a query point $q \in F \setminus C_\emptyset$. We find the farthest sites from q in the other three maps using their query data structures.

By Corollary 3, our query problem reduces to the vertical ray shooting queries. We use the data structure by Giora and Kaplan [15] for vertical ray shooting queries on $O(n + m)$ horizontal line segments in Q_{y^+} , which requires $O((n + m) \log(n + m))$ time and $O(n + m)$ space for construction. Let h be the horizontal segment in Q_{y^+} hit first by the vertical ray emanating from q going downwards. We can find h in $O(\log(n + m))$ time using the ray shooting structure. If no horizontal segment in Q_{y^+} is hit by the ray, q is y^+ -reachable from no site. Otherwise, there are $O(m)$ boundary points on h , sorted in increasing order of x -coordinate. With those boundary points, we can find $f(q)$ for a query point q in $O(\log m)$ time using binary search. Thus, a farthest-neighbor query takes $O(\log(n + m))$ time in total.

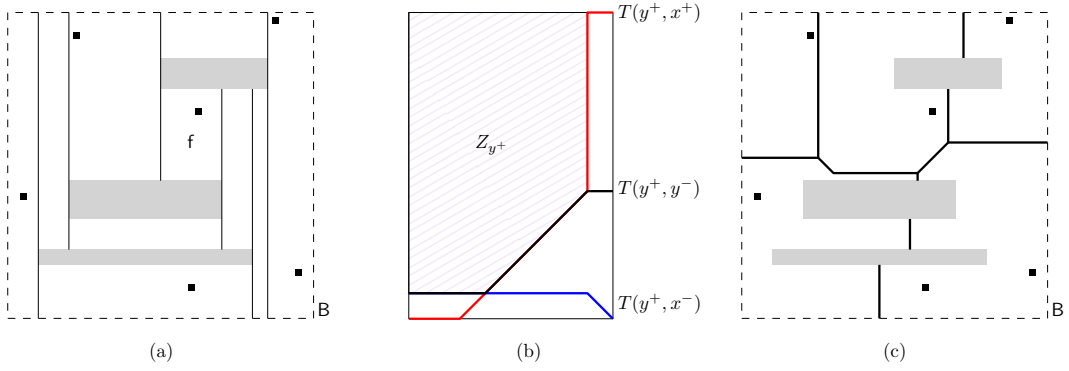
Once the farthest sites of q for each of the four data structures is found, we take the sites with the largest distance among them as the farthest sites $f(q)$ of S from q . Combining Lemmas 6, 7 and 8 with query time, we have the following theorem.

► **Theorem 9.** *We can construct a data structure for m point sites in the presence of n axis-aligned rectangular obstacles in the plane in $O(nm + n \log n + m \log m)$ time and $O(nm)$ space that answers any L_1 farthest-neighbor query in $O(\log(n + m))$ time.*

5 Computing the Explicit Farthest-point Voronoi Diagram

We construct the explicit farthest-point Voronoi diagram $FVD = FVD(S, R)$ of a set S of m point sites in the presence of a set R of n rectangular obstacles in the plane. It is known that FVD requires $\Omega(nm)$ space [5, 7]. It takes $\Omega(n \log n)$ time to compute the geodesic distance between two points in F [12]. By a reduction from the sorting problem, it can be shown to take $\Omega(m \log m)$ time for computing the farthest-point Voronoi diagram of m point sites in the plane. We present an $O(nm + n \log n + m \log m)$ -time algorithm using $O(nm)$ space that matches the time and space lower bounds. This is the first optimal algorithm for constructing the farthest-point Voronoi diagram of points in the presence of obstacles in the plane in both time and space.

We construct Q_{y^+} using the plane sweep in Section 4. During the plane sweep, we find all horizontal edges of M_{y^+} and insert them into Q_{y^+} as segments. We find all the lower endpoints of the vertical edges of M_{y^+} and insert them as boundary points in \mathcal{B} . We also find the upper endpoints of vertical edges of M_{y^+} . By connecting those endpoints using vertical segments appropriately, we can construct M_{y^+} from Q_{y^+} in a doubly connected edge list without increasing the time and space complexities. The other three maps can also be constructed in the same way in the same time and space.



■ **Figure 6** (a) Vertical decomposition F_V . f is a face of F_V . (b) Z_{y^+} is a region in f above the upper envelope of three traces, $T(y^+, y^-)$, $T(y^+, x^+)$ and $T(y^+, x^-)$. (c) Explicit geodesic L_1 farthest-point Voronoi diagram FVD.

We construct the farthest-point Voronoi diagram FVD using the four maps explicitly. Note that $f(p) = f_{y^+}(p)$ for any point p lying on the top side of B . Thus, it suffices to compute FVD in $F \cap B$. For ease of description, we assume that the x -coordinates of the rectangles in R are all distinct. We consider a vertical decomposition F_V obtained by drawing maximal vertical line segments contained in $F \cap B$ of which each is extended from a vertical side of a hole of F . Let V be a set of such vertical line segments. $F \setminus \bigcup_{\ell \in V} \ell$ consists of $O(n)$ connected faces. Each face is a rectangle since each hole of F is a rectangle and F is bounded by B . See Figure 6(a).

Any two farthest-point maps M_1, M_2 have a *bisector* which consists of the points in F having the same distance to their farthest sites in M_1 and in M_2 . The four maps define six bisectors. In a face of F_V , the six bisectors and some axis-aligned segments partition the face into *zones* such that FVD restricted to one zone coincides with the diagram in the corresponding region of a farthest-point map. Thus, we compute the bisectors between maps in each face of F_V , partition the face into zones, find the region of a farthest-point map corresponding to each zone, and then glue the regions and faces to compute FVD completely.

5.1 Bisectors of farthest-point maps

We define the *bisector* between M_δ and $M_{\delta'}$ as $B(\delta, \delta') = \{q \in F \mid d_\delta(q) = d_{\delta'}(q)\}$ for any two distinct $\delta, \delta' \in \{y^+, y^-, x^+, x^-\}$. We show that any vertical line intersects $B(y^+, y^-)$ in at most one point, and any vertical line segment contained in F intersects $B(y^+, x^+)$ (and $B(y^+, x^-)$) in at most one connected component. Thus, these three bisectors contained in a face of F_V are x -monotone. Details are given in the full version.

For each face f of F_V , we compute the portion of $B(y^+, y^-)$ contained in f . As $B(y^+, y^-) \cap f$ is x -monotone, we sweep a vertical line L from $x_1(f)$ to $x_2(f)$ maintaining a point $p \in f \cap L$ with $d_{y^+}(p) = d_{y^-}(p)$. First, we compute p lying on the left side of f as follows. There are $O(m)$ intersections of the left side of f with the horizontal segments of Q_{y^+} and Q_{y^-} as any vertical line segment contained in F intersects $O(m)$ horizontal segments of them. For each intersection point q , we compute $d_{y^+}(q)$ and $d_{y^-}(q)$, and find two consecutive points q_1 and q_2 among the intersection points by y -coordinate such that $d_{y^+}(q_1) \leq d_{y^-}(q_1)$ and $d_{y^-}(q_2) \leq d_{y^+}(q_2)$. We can compute q_1 and q_2 in $O(m)$ time using Q_{y^+} and Q_{y^-} . Then we compute p lying on $q_1 q_2$.

Having the distance functions, we have the slope of the bisector incident to p . Let $\vec{\ell}$ be the half-line from p with the slope going rightward. We find the first point p' on $\vec{\ell}$ from p at which the slope of $d_{y^+}(p')$ or $d_{y^-}(p')$ changes. Since the slope of $d_{y^+}(p')$ changes at most once within a cell of M_{y^+} , we can find p' in time linear to the complexity of the cells containing p of the maps. If there are two or more such points, p is the point with the maximum y -coordinate among them.

There may be no point p satisfying $d_{y^+}(p) = d_{y^-}(p)$ if there is a point $q \in f \cap L$ such that $d_{y^+}(q') > d_{y^-}(q')$ for every point q' lying above q , and $d_{y^+}(q') < d_{y^-}(q')$ for every point q' lying below q . We maintain the point q in this case. Note that q follows a horizontal segment during the plane sweep, and thus we can find the first point p with $d_{y^+}(p) = d_{y^-}(p)$ using a horizontal half-line from q .

During the plane sweep, p or q moves along $B(y^+, y^-)$ rightwards until it meets the right side of f . We compute the other bisectors in f similarly.

We compute the trace $T(y^+, y^-)$ of p and q during the sweep. Observe that every vertical line intersecting f also intersects the trace in one point t . Moreover, if the line intersects $B(y^+, y^-) \cap f$, t is the topmost point of the intersection. Since we have M_{x^+} and M_{x^-} , we can compute the two traces $T(y^+, x^+)$ and $T(y^+, x^-)$ similarly.

We observe that each bisector and trace in f has $O(m)$ complexity. We get the distance functions using Q_{y^+} , Q_{y^-} , Q_{x^+} , and Q_{x^-} which consist of $O(n + m)$ line segments and support $O(\log(n + m))$ query time. After computing those distance functions, the traces can be constructed in time linear to their complexities. Thus, in total it takes $O(nm + n \log n + m \log m)$ time to construct the traces for all faces.

5.2 Partitioning f into zones

With the three traces $T(y^+, y^-)$, $T(y^+, x^+)$, $T(y^+, x^-)$ in f , we compute the zone Z_{y^+} in f corresponding to M_{y^+} in f . Let T be an upper envelope of $T(y^+, y^-)$, $T(y^+, x^+)$ and $T(y^+, x^-)$. Then Z_{y^+} is the set of points lying above T in f . See Figure 6(b). The following lemma can be shown using the lemmas in the full version.

► **Lemma 10.** *For any point $p \in Z_{y^+}$, $f(p) = f_{y^+}(p)$.*

Similarly, we define the other three zones Z_{y^-} , Z_{x^+} , and Z_{x^-} . Note that $d_\delta(p) > d_{\delta'}(p)$ for every point $p \in Z_\delta$ for distinct $\delta, \delta' \in \{y^+, y^-, x^+, x^-\}$. By Lemma 10, $FVD \cap Z_{y^+}$ coincides with M_{y^+} . We copy the corresponding farthest-point map of δ into Z_δ for each $\delta \in \{y^+, y^-, x^+, x^-\}$.

We call $f \setminus (Z_{y^+} \cup Z_{y^-} \cup Z_{x^+} \cup Z_{x^-})$ the bisector zone. Every point p in the bisector zone lies on a bisector of two or more maps. Thus, for each bisector of two maps, we copy one of the maps into the corresponding zone.

5.3 Gluing along boundaries

We first glue the zones along their boundaries in each face of F_V . For each edge e incident to two zones, we check whether the two cells incident to the edge have the same farthest site or not. If they have the same farthest site, e is not a Voronoi edge of FVD. Then we remove the edge and merge the cells into one. If they have different farthest sites, e is a Voronoi edge of FVD. This takes $O(nm)$ time in total, which is linear to the number of Voronoi edges and cells in FVD.

After gluing zones in every face, we glue the faces of F_V along their boundaries. Since e is a vertical line segment and incident to more than two cells, we divide e into pieces such that any point in the same piece e' is incident to the same set of two cells. If both cells incident

to e' have the same farthest site, e' is not a Voronoi edge of FVD. Then we remove the edge and merge the cells. If they have different farthest sites, e' is a Voronoi edge of FVD. There are $O(n)$ vertical line segments in V and each of them intersects $O(m)$ cells of FVD, so it takes $O(nm)$ time in total. Then we obtain the geodesic L_1 farthest-point Voronoi diagram FVD explicitly. See Figure 6(c).

► **Theorem 11.** *We can compute the L_1 farthest-point Voronoi diagram of m point sites in the presence of n axis-aligned rectangular obstacles in the plane in $O(nm + n \log n + m \log m)$ time and $O(nm)$ space.*

► **Corollary 12.** *We can compute the L_1 geodesic center of m point sites in the presence of n axis-aligned rectangular obstacles in the plane in $O(nm + n \log n + m \log m)$ time and $O(nm)$ space.*

6 Concluding Remarks

We present an optimal algorithm for computing the farthest-point Voronoi diagram of point sites in the presence of rectangular obstacles. However, our algorithm may not work for more general obstacles as it is, because some properties we use for the axis-aligned rectangles including their convexity may not hold any longer. Our results, however, may serve as a stepping stone to closing the gap to the optimal bounds.

References

- 1 A. Aggarwal, L.J. Guibas, J. Saxe, and P.W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete & Computational Geometry*, 4(6):591–604, 1989.
- 2 H. Alt, O. Cheong, and A. Vigneron. The Voronoi diagram of curved objects. *Discrete & Computational Geometry*, 34(3):439–453, 2005.
- 3 B. Aronov. On the geodesic Voronoi diagram of point sites in a simple polygon. *Algorithmica*, 4(1):109–140, 1989.
- 4 B. Aronov, S. Fortune, and G. Wilfong. The furthest-site geodesic Voronoi diagram. *Discrete & Computational Geometry*, 9(3):217–255, 1993.
- 5 S.W. Bae and K.-Y. Chwa. The geodesic farthest-site Voronoi diagram in a polygonal domain with holes. In *Proceedings of the 25th Annual Symposium on Computational Geometry (SoCG)*, pages 198–207, 2009.
- 6 B. Ben-Moshe, B.K. Bhattacharya, and Q. Shi. Farthest neighbor Voronoi diagram in the presence of rectangular obstacles. In *Proceedings of the 13th Canadian Conference on Computational Geometry (CCCG)*, pages 243–246, 2005.
- 7 B. Ben-Moshe, M.J. Katz, and J.S.B. Mitchell. Farthest neighbors and center points in the presence of rectangular obstacles. In *Proceedings of the 17th Annual Symposium on Computational Geometry (SoCG)*, pages 164–171, 2001.
- 8 O. Cheong, H. Everett, M. Glisse, J. Gudmundsson, S. Hornus, S. Lazard, M. Lee, and H.-S. Na. Farthest-polygon Voronoi diagrams. *Computational Geometry*, 44(4):234–247, 2011.
- 9 L.P. Chew and R.L. Dyrsdale III. Voronoi diagrams based on convex distance functions. In *Proceedings of the 1st annual symposium on Computational geometry (SoCG)*, pages 235–244, 1985.
- 10 J. Choi, C.-S. Shin, and S.K. Kim. Computing weighted rectilinear median and center set in the presence of obstacles. In *International Symposium on Algorithms and Computation*, pages 30–40. Springer, 1998.
- 11 J. Choi and C. Yap. Monotonicity of rectilinear geodesics in d -space. In *Proceedings of the 12th Annual Symposium on Computational Geometry (SoCG)*, pages 339–348, 1996.

- 12 P.J. De Rezende, D.-T. Lee, and Y.-F. Wu. Rectilinear shortest paths with rectangular barriers. In *Proceedings of the 1st Annual Symposium on Computational Geometry (SoCG)*, pages 204–213, 1985.
- 13 H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. *Discrete & Computational Geometry*, 1(1):25–44, 1986.
- 14 S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(1):153–174, 1987.
- 15 Y. Giora and H. Kaplan. Optimal dynamic vertical ray shooting in rectilinear planar subdivisions. *ACM Transactions on Algorithms*, 5(3):28:1–51, 2009.
- 16 J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.
- 17 R. Klein. Abstract Voronoi diagrams and their applications. In *Proceedings of the 4th International Workshop on Computational Geometry (EuroCG)*, pages 148–157. Springer, 1988.
- 18 D.-T. Lee. Two-dimensional Voronoi diagrams in the L_p -metric. *Journal of the ACM*, 27(4):604–618, 1980.
- 19 E. Oh. Optimal algorithm for geodesic nearest-point Voronoi diagrams in simple polygons. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 391–409, 2019.
- 20 E. Oh and H.-K. Ahn. Voronoi diagrams for a moderate-sized point-set in a simple polygon. *Discrete & Computational Geometry*, 63(2):418–454, 2020.
- 21 E. Oh, L. Barba, and H.-K. Ahn. The geodesic farthest-point Voronoi diagram in a simple polygon. *Algorithmica*, 82(5):1434–1473, 2020.
- 22 E. Papadopoulou and S.K. Dey. On the farthest line-segment Voronoi diagram. *International Journal of Computational Geometry & Applications*, 23(06):443–459, 2013.
- 23 E. Papadopoulou and D.T. Lee. The L_∞ Voronoi diagram of segments and VLSI applications. *International Journal of Computational Geometry & Applications*, 11(05):503–528, 2001.
- 24 M.I. Shamos and D. Hoey. Closest-point problems. In *Proceedings of the 16th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 151–162, 1975.
- 25 H. Wang. An optimal deterministic algorithm for geodesic farthest-point Voronoi diagrams in simple polygons. In *Proceedings of the 37th International Symposium on Computational Geometry (SoCG)*, pages 59:1–59:15, 2021.

Point Separation and Obstacle Removal by Finding and Hitting Odd Cycles

Neeraj Kumar ✉

Department of Computer Science, University of California, Santa Barbara, CA, USA

Daniel Lokshantov ✉

Department of Computer Science, University of California, Santa Barbara, CA, USA

Saket Saurabh ✉

Institute of Mathematical Sciences, Chennai, India

University of Bergen, Norway

Subhash Suri ✉

Department of Computer Science, University of California, Santa Barbara, CA, USA

Jie Xue ✉

New York University Shanghai, China

Abstract

Suppose we are given a pair of points s, t and a set \mathcal{S} of n geometric objects in the plane, called obstacles. We show that in polynomial time one can construct an auxiliary (multi-)graph G with vertex set \mathcal{S} and every edge labeled from $\{0, 1\}$, such that a set $\mathcal{S}_d \subseteq \mathcal{S}$ of obstacles separates s from t if and only if $G[\mathcal{S}_d]$ contains a cycle whose sum of labels is odd. Using this structural characterization of separating sets of obstacles we obtain the following algorithmic results.

In the OBSTACLE-REMOVAL problem the task is to find a curve in the plane connecting s to t intersecting at most q obstacles. We give a $2.3146^q n^{O(1)}$ algorithm for OBSTACLE-REMOVAL, significantly improving upon the previously best known $q^{O(q^3)} n^{O(1)}$ algorithm of Eiben and Lokshantov (SoCG'20). We also obtain an alternative proof of a constant factor approximation algorithm for OBSTACLE-REMOVAL, substantially simplifying the arguments of Kumar et al. (SODA'21).

In the GENERALIZED POINTS-SEPARATION problem input consists of the set \mathcal{S} of obstacles, a point set A of k points and p pairs $(s_1, t_1), \dots, (s_p, t_p)$ of points from A . The task is to find a minimum subset $\mathcal{S}_r \subseteq \mathcal{S}$ such that for every i , every curve from s_i to t_i intersects at least one obstacle in \mathcal{S}_r . We obtain $2^{O(p)} n^{O(k)}$ -time algorithm for GENERALIZED POINTS-SEPARATION. This resolves an open problem of Cabello and Giannopoulos (SoCG'13), who asked about the existence of such an algorithm for the special case where $(s_1, t_1), \dots, (s_p, t_p)$ contains all the pairs of points in A . Finally, we improve the running time of our algorithm to $f(p, k) \cdot n^{O(\sqrt{k})}$ when the obstacles are unit disks, where $f(p, k) = 2^{O(p)} k^{O(k)}$, and show that, assuming the Exponential Time Hypothesis (ETH), the running time dependence on k of our algorithms is essentially optimal.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases points-separation, min color path, constraint removal, barrier resilience

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.52

Related Version *Full Version:* <https://arxiv.org/abs/2203.08193>

Funding *Daniel Lokshantov:* BSF award 2018302 and NSF award CCF-2008838

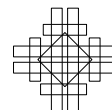
Saket Saurabh: European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416), and Swarnajayanti Fellowship (No. DST/SJF/MSA01/2017-18).

Subhash Suri: NSF award CCF-1814172



© Neeraj Kumar, Daniel Lokshantov, Saket Saurabh, Subhash Suri, and Jie Xue; licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 52; pp. 52:1–52:14
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Suppose we are given a set \mathcal{S} of geometric objects in the plane, and we want to modify \mathcal{S} in order to achieve certain guarantees on coverage of paths between a given set A of points. Such problems have received significant interest in sensor networks [2, 4, 6, 16], robotics [9, 12] and computational geometry [3, 8, 11]. There have been two closely related lines of work on this topic: (i) *remove* a smallest number of obstacles from \mathcal{S} to satisfy *reachability* requirements for points in A , and (ii) *retain* a smallest number of obstacles to satisfy *separation* requirements for points in A .

In the most basic version of these problems the set A consists of just two points s and t . Specifically, in OBSTACLE-REMOVAL the task is to find a smallest possible set $\mathcal{S}_d \subseteq \mathcal{S}$ such that there is a curve from s to t in the plane avoiding all obstacles in $\mathcal{S} \setminus \mathcal{S}_d$. In 2-POINTS-SEPARATION the task is to find a smallest set $\mathcal{S}_r \subseteq \mathcal{S}$ such that every curve from s to t in the plane intersects at least one obstacle in \mathcal{S}_r . It is quite natural to require the obstacles in the set \mathcal{S} to be connected. Indeed, removing the connectivity requirements results in problems that are computationally intractable [8, 10, 21].

When the obstacles are required to be connected OBSTACLE-REMOVAL remains NP-hard, but becomes more tractable from the perspective of approximation algorithms and parameterized algorithms. For approximation algorithms, Bereg and Kirkpatrick [4] designed a constant factor approximation for unit disk obstacles. Chan and Kirkpatrick [6, 7] improved the approximation factor for unit disk obstacles. Korman et al. [14] obtained a $(1 + \epsilon)$ -approximation algorithm for the case when obstacles are fat, similarly sized, and no point in the plane is contained in more than a constant number of obstacles. Whether a constant factor approximation exists for general obstacles was posed repeatedly as an open problem [3, 6, 7] before it was resolved in the affirmative by a subset of the authors of this article [21].

For parameterized algorithms, Korman et al. [14] designed an algorithm for OBSTACLE-REMOVAL with running time $f(q)n^{O(1)}$ for determining whether there exists a solution \mathcal{S}_d of size at most q , when obstacles are fat, similarly sized, and no point in the plane is contained in more than a constant number of obstacles. Eiben and Kanj [8, 10] generalized the result of Korman et al. [14], and posed as an open problem the existence of a $f(q)n^{O(1)}$ time algorithm for OBSTACLE-REMOVAL with general connected obstacles. Eiben and Lokshtanov [11] resolved this problem in the affirmative, providing an algorithm with running time $q^{O(q^3)}n^{O(1)}$.

Like OBSTACLE-REMOVAL, the 2-POINTS-SEPARATION problem becomes more tractable when the obstacles are connected. Cabello and Giannopoulos [5] showed that 2-POINTS-SEPARATION with connected obstacles is polynomial time solvable. They show that the more general POINTS-SEPARATION problem where we are given a point set A and asked to find a minimum size set $\mathcal{S}_r \subseteq \mathcal{S}$ that separates every pair of points in A , is NP-complete, even when all obstacles are unit disks. They leave as an open problem to determine the existence of $f(k)n^{O(1)}$ and $f(k)n^{g(k)}$ time algorithms for POINTS-SEPARATION, where $k = |A|$.

Our Results and Techniques

Our main result is a structural characterization of separating sets of obstacles in terms of odd cycles in an auxiliary graph.

► **Theorem 1.** *There exists a polynomial time algorithm that takes as input a set \mathcal{S} of obstacles in the plane, two points s and t , and outputs a (multi-)graph G with vertex set \mathcal{S} and every edge labeled from $\{0, 1\}$, such that a set $\mathcal{S}_d \subseteq \mathcal{S}$ of obstacles separates s from t if and only if $G[\mathcal{S}_d]$ contains a cycle whose sum of labels is odd.*

The proof of Theorem 1 is an application of the well known fact that a closed curve separates s from t if and only if it crosses a curve from s to t an odd number of times. Theorem 1 allows us to re-prove, improve, and generalize a number of results for OBSTACLE-REMOVAL, 2-POINTS-SEPARATION and POINTS-SEPARATION in a remarkably simple way. More concretely, we obtain the following results.

- *There exists a polynomial time algorithm for 2-POINTS-SEPARATION.*

Here is the proof: construct the graph G from Theorem 1 and find the shortest odd cycle, which is easy to do in polynomial time. This re-proves the main result of Cabello and Giannopoulos [5]. Next we turn to OBSTACLE-REMOVAL, and obtain an improved parameterized algorithm and simplified approximation algorithms.

- *There exists an algorithm for OBSTACLE-REMOVAL that determines whether there exists a solution size set \mathcal{S} of size at most q in time $2.3146^q n^{O(1)}$.*

Here is a proof sketch: construct the graph G from Theorem 1 and determine whether there exists a subset \mathcal{S}_d of \mathcal{S} of size at most q such that $G - \mathcal{S}_d$ does not have any odd label cycle. This can be done in time $2.3146^q n^{O(1)}$ using the algorithm of Lokshtanov et al. [18] for ODD CYCLE TRANSVERSAL.¹ This parameterized algorithm improves over the previously best known parameterized algorithm for OBSTACLE-REMOVAL of Eiben and Lokshtanov [11] with running time $q^{O(q^3)} n^{O(1)}$.

If we run an approximation algorithm for ODD CYCLE TRANSVERSAL on G instead of a parameterized algorithm, we immediately obtain an approximation algorithm for OBSTACLE-REMOVAL with the same ratio. Thus, the $O(\sqrt{\log n})$ -approximation algorithm for ODD CYCLE TRANSVERSAL [1, 15] implies a $O(\sqrt{\log n})$ -approximation algorithm for OBSTACLE-REMOVAL as well. Going a little deeper we observe that the structure of G implies that the standard Linear Programming relaxation of ODD CYCLE TRANSVERSAL on G only has a constant integrality gap. This yields a constant factor approximation for OBSTACLE-REMOVAL, substantially simplifying the approximation algorithm of Kumar et al [21].

- *There exists a constant factor approximation for OBSTACLE-REMOVAL.*

Finally we turn our attention back to a generalization of POINTS-SEPARATION, called GENERALIZED POINTS-SEPARATION. Here, instead of separating all k points in A from each other, we are only required to separate p specific pairs $(s_1, t_1), \dots, (s_p, t_p)$ of points in A (which are specified in the input). We apply Theorem 1 several times, each time with the same obstacle set \mathcal{S} , but with a different pair (s_i, t_i) . Let G_i be the graph resulting from the construction with the pair (s_i, t_i) . Finding a minimum size set \mathcal{S}_r of obstacles that separates s_i from t_i for every i now amounts to finding a minimum size set \mathcal{S}_r such that $G_i[\mathcal{S}_r]$ contains an odd label cycle for every i . The graph in the construction of Theorem 1 does not depend on the points (s_i, t_i) - only the labels of the edges do. Thus G_1, \dots, G_p are copies of the same graph G , but with p different edge labelings. Our task now is to find a subgraph of G on the minimum number of vertices, such that the subgraph contains an odd labeled cycle with respect to each one of the p labels. We show that such a subgraph has at most $O(p)$ vertices of degree at least 3 and use this to obtain a $2^{O(p^2)} n^{O(p)}$ time algorithm

¹ The only reason this is a proof sketch rather than a proof is that the algorithm of Lokshtanov et al. [18] works for unlabeled graphs, while G has edges with labels 0 or 1. This difference can be worked out using a well-known and simple trick of subdividing every edge with label 0 (see Section 4).

for GENERALIZED POINTS-SEPARATION. This implies a $2^{O(k^4)}n^{O(k^2)}$ time algorithm for POINTS-SEPARATION, resolving the open problem of Cabello and Giannopoulos [5]. With additional technical effort we are able to bring down the running time of our algorithm for GENERALIZED POINTS-SEPARATION to $2^{O(p)}n^{O(k)}$. This turns out to be close to the best one can do. On the other hand, for *pseudo-disk* obstacles we can get a faster algorithm.

- There exists a $2^{O(p)}n^{O(k)}$ time algorithm for GENERALIZED POINTS-SEPARATION, and a $n^{O(\sqrt{k})}$ time algorithm for GENERALIZED POINTS-SEPARATION with *pseudo-disk* obstacles.
- A $f(k)n^{o(k/\log k)}$ time algorithm for POINTS-SEPARATION, or a $f(k)n^{o(\sqrt{k})}$ time algorithm for POINTS-SEPARATION with *pseudo-disk* obstacles would violate the ETH [13].

2 Preliminaries

All graphs used in this paper are undirected. It will also be more convenient to sometimes consider multi-graphs, in which self-loops and parallel edges are allowed. The *degree* of a vertex is the number of adjacent edges.

The *arrangement* $\text{Arr}(\mathcal{S})$ of a set of obstacles \mathcal{S} is a subdivision of the plane induced by the boundaries of the obstacles in \mathcal{S} . The faces of $\text{Arr}(\mathcal{S})$ are connected regions and edges are parts of obstacle boundaries. The *arrangement graph* $G_{\text{Arr}} = (V, E)$ is the dual graph of the arrangement whose vertices are faces of $\text{Arr}(\mathcal{S})$ and edges connect neighboring faces. The complexity of the arrangement is the size of its arrangement graph which we denote by $|\text{Arr}(\mathcal{S})|$. We assume that the size of the arrangement is polynomial in the number of obstacles, that is $|\text{Arr}(\mathcal{S})| = |G_{\text{Arr}}| = n^{O(1)}$. This is indeed true for most reasonable obstacle models such as polygons or low-degree splines.

Plane curves and Crossings. A *plane curve* (or simply *curve*) is specified by a continuous function $\pi : [0, 1] \rightarrow \mathbb{R}^2$, where the points $\pi(0)$ and $\pi(1)$ are called the *endpoints* (we also use the notation π to denote the image of the path function π). A curve is *simple* if it is injective, and is *closed* if its two endpoints are the same. We say a curve π *separates* a pair (a, b) of two points in \mathbb{R}^2 if a and b belong to different connected components of $\mathbb{R}^2 \setminus \pi$.

A *crossing* of π with π' is an element of the set $\{t \in [0, 1] \mid \pi(t) \in \pi'\}$. We will often be concerned with the *number* of times π crosses π' . This is defined as $|\{t \in [0, 1] \mid \pi(t) \in \pi'\}|$. Whenever we count the number of times a curve π crosses another curve π' we shall assume that (and ensure that) $|\{t \in [0, 1] \mid \pi(t) \in \pi'\}|$ is finite and that π and π' are *transverse*. That is for every $t \in [0, 1]$ such that $\pi(t) \in \pi'$ there exists an $\epsilon > 0$ such that the intersection of $\pi \cup \pi'$ with an ϵ radius ball around $\pi(t)$ is homotopic with two orthogonal lines. We will make frequent use of the following basic topological fact.

► **Fact 2.** Let π be a curve with endpoints $a, b \in \mathbb{R}^2$. We have that (i) A simple closed curve γ separates (a, b) iff π crosses γ an odd number of times. (ii) If π crosses a closed curve γ an odd number of times, then γ separates (a, b) .

3 Labeled Intersection Graph of Obstacles

We begin by describing the construction of the labeled intersection graph $G_{\mathcal{S}} = (\mathcal{S}, X)$ of the obstacles \mathcal{S} . For the ease of exposition, we will use S to refer to the obstacle $S \in \mathcal{S}$ as well as the vertex for S in $G_{\mathcal{S}}$ interchangeably.

Constructing the graph G_S . For every obstacle $S \in \mathcal{S}$ we first select an arbitrary point $\text{ref}(S) \in S$ and designate it to be the *reference point* of the obstacle. Next, we select the *reference curve* π to be a simple curve in the plane connecting s and t such that including it to the arrangement $\text{Arr}(\mathcal{S})$ does not significantly increase its complexity. That is, we want to ensure that $|\text{Arr}(\mathcal{S} \cup \pi)| = O(|\text{Arr}(\mathcal{S})|)$. Additionally, the reference curve π is chosen such that there exists an $\epsilon > 0$ and π is disjoint from an ϵ ball around every intersection point of two obstacles in $\text{Arr}(\mathcal{S})$ and from an ϵ ball around every reference point $\text{ref}(S)$ for $S \in \mathcal{S}$.

As long as the intersection of every pair of obstacles is finite and their arrangement has bounded size, a suitable choice for π always exists (and can be efficiently computed). For example one can choose π to be the plane curve corresponding to an s - t path in G_{Arr} .

We will now add edges to G_S as follows. (See also Figure 1(c) for an example.)

- For every obstacle $S \in \mathcal{S}$ that contains s or t , add a self-loop $e = (S, S)$ with $\text{lab}(e) = 1$.
- For every pair of obstacles $S, S' \in \mathcal{S}$ that intersect, we add edges to G as follows.
 - Add an edge $e_0 = (S, S')$ with $\text{lab}(e_0) = 0$ if there exists a curve connecting $\text{ref}(S)$ and $\text{ref}(S')$ contained in the region $S \cup S'$ that crosses π an *even* number of times.
 - Add an edge $e_1 = (S, S')$ with $\text{lab}(e_1) = 1$ if there exists a curve connecting $\text{ref}(S)$ and $\text{ref}(S')$ contained in the region $S \cup S'$ that crosses π an *odd* number of times.

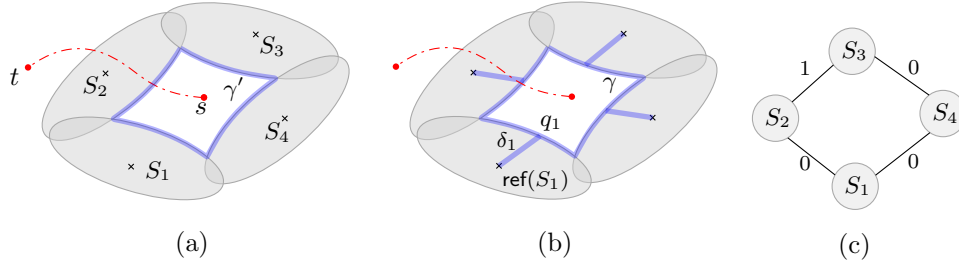
Checking whether there exists a curve contained in the region $S \cup S'$ with endpoints $\text{ref}(S)$ and $\text{ref}(S')$ that crosses π an odd (resp. even) number of times can be done in time linear in the size of arrangement $\text{Arr}' = \text{Arr}(S \cup S' \cup \pi)$. Specifically, we build the arrangement graph $G_{\text{Arr}'}$ and only retain edges (f_i, f_j) such that the faces $f_i, f_j \in S \cup S'$. If the common boundary of faces f_i, f_j is a portion of π , we assign a label 1 to the edge (f_i, f_j) , otherwise we assign it a label 0. An odd (resp. even) labeled walk in $G_{\text{Arr}'}$ connecting the faces containing $\text{ref}(S)$ and $\text{ref}(S')$ gives us the desired plane curve π_{ij} . Since edges of $G_{\text{Arr}'}$ connect adjacent faces of Arr' , we can ensure that the intersections between curve π_{ij} and the edges of arrangement (including parts of reference curve π) are all transverse.

We are now ready to prove the following important structural property of the graph G_S .

► **Lemma 3.** *A set of obstacles $\mathcal{S}' \subseteq \mathcal{S}$ in the graph G_S separates the points s and t if and only if the induced graph $H = G_S[\mathcal{S}']$ contains an odd labeled cycle.*

Proof. (\Rightarrow) For the forward direction, suppose we are given a set of obstacles \mathcal{S}' that separate s from t . If s or t are contained in some obstacle, then we must have an odd self-loop in G_S and we will be done. Otherwise, assume that s, t lie in the exterior of all obstacles, so we have $s, t \notin \mathcal{R}(\mathcal{S}')$ where $\mathcal{R}(\mathcal{S}') = \bigcup_{S \in \mathcal{S}'} S$ is the region bounded by obstacles in \mathcal{S}' . Observe that s, t must lie in different connected regions R_s, R_t of $\mathbb{R}^2 \setminus \mathcal{R}(\mathcal{S}')$ or else the set \mathcal{S}' would not separate them. At least one of R_s or R_t must be bounded, wlog assume it is R_s . Let γ' be the simple closed curve that is the common boundary of $\mathcal{R}(\mathcal{S}')$ and R_s . We have that γ' encloses s but not t and therefore separates s from t . Using first statement of Fact 2, we obtain that γ' crosses the reference curve π an odd number of times. Observe that the curve γ' consists of multiple *sections* $\alpha'_1 \rightarrow \alpha'_2 \cdots \rightarrow \alpha'_r$ where each curve α'_i is part of the boundary of some obstacle S_i . For each of these curves α'_i , we add a *detour* to and back from the reference point $\text{ref}(S_i)$ of the obstacle it belongs. Specifically, let q_i be an arbitrary point on the curve α'_i and let $\alpha'_{i\ell}, \alpha'_{ir}$ be the portion of α'_i before and after q_i respectively. We add the *detour curve* $\delta_i = q_i \rightarrow \text{ref}(S_i) \rightarrow q_i$ ensuring that it always stays within the obstacle S_i which is possible because the obstacles are connected. (Same as before the curve δ_i can be chosen to be transverse with π by considering the corresponding walk in graph of $\text{Arr}(S_i \cup \pi)$.) Let $\alpha_i = \alpha'_{i\ell} \rightarrow \delta_i \rightarrow \alpha'_{ir}$ be the curve obtained by adding detour δ_i to α'_i . Let

$\gamma = \alpha_1 \rightarrow \alpha_2 \cdots \rightarrow \alpha_r$ be the closed curve obtained by adding these detours to γ' . Note that γ is not necessarily simple as the detour curves may intersect each other. Every detour δ_i consists of identical copies of two curves, so it crosses the reference curve π an even number of times. Since γ' crosses π an odd number of times, the curve γ also crosses π an odd number of times. (See also Figure 1.) Observe that γ and γ' are transverse with π because intersections of π and obstacle boundaries are transverse and the detour curves δ_i are chosen to be transverse with π .



■ **Figure 1** (a) The curve γ' shown shaded in blue is the common boundary of $\mathcal{R}(S')$ and region R_s (b) Adding detours δ_i to obtain curve γ (c) Labeled Intersection graph G_S ob obstacles.

We will now translate the curve γ to a *walk* in the labeled intersection graph G_S . Specifically, consider the section of γ between two consecutive detours: $\gamma_{i,i+1} = \text{ref}(S_i) \rightarrow q_i \rightarrow q_{i+1} \rightarrow \text{ref}(S_{i+1})$. Therefore the obstacles S_i, S_{i+1} must intersect and we have a curve $\gamma_{i,i+1}$ connecting their reference points contained in the region $S_i \cup S_{i+1}$ that also intersects the reference curve π an odd (resp. even) number of times. By construction, G_S must contain an edge $e_{i,i+1}$ with label 1 (resp. 0). By replacing all these sections of γ with the corresponding edges of G_S , we obtain an odd-labeled closed walk W in G_S . Of all the odd-labeled closed sub-walks of W , we select one that is inclusion minimal. This gives a simple odd-labeled cycle in $G_S[S']$.

(\Leftarrow) The reverse direction is relatively simpler. Given an odd-labeled cycle in $G_S[S']$, we obtain a closed curve γ in the plane contained in region $\mathcal{R}(S')$ as follows. For every edge $e_i = (S, S')$ of the cycle with label $\text{lab}(e_i)$, we consider the curve γ_i that connects the reference points $\text{ref}(S)$ and $\text{ref}(S')$ contained in $S \cup S'$ and crosses the reference curve π consistent with $\text{lab}(e_i)$. Moreover γ_i needs to be transverse with π . Such a curve exists by construction of G_S . Combining these curves γ_i in order gives us a closed curve γ in the plane that crosses π an odd number of times. Although this curve may be self intersecting, from second statement of Fact 2, we have that γ separates s and t . \blacktriangleleft

The construction of the graph G_S , together with Lemma 3 prove Theorem 1.

2-Points-separation as Shortest Odd Cycle in G_S . From Lemma 3, it follows that a minimum set of obstacles that separates s from t corresponds to an odd-labeled cycle in G_S with fewest vertices. This readily gives a polytime algorithm for 2-POINTS-SEPARATION. In particular, for a fixed starting vertex, we can compute the shortest odd cycle in G_S in $O(|S|^2)$ time by the following well-known technique. Consider an unlabeled auxiliary graph G' with vertex set is $S \times \{0, 1\}$. For every edge $e = (S, S')$ of G_S , we add edges $\{(S, 0), (S', 0)\}$ and $\{(S, 1), (S', 1)\}$ if $\text{lab}(e) = 0$. Otherwise, we add the edges $\{(S, 0), (S', 1)\}$ and $\{(S, 1), (S', 0)\}$. The shortest odd cycle containing a fixed vertex S is the shortest path in G' between vertices $(S, 0)$ and $(S, 1)$. Repeating over all starting vertices gives the shortest odd cycle in G_S . This

can be easily extended for the node-weighted case which gives us the following useful lemma that also yields a polynomial time algorithm for 2-POINTS-SEPARATION, improving a result of Cabello and Giannopoulos [5].

► **Lemma 4.** *There exists a polynomial time algorithm for computing a minimum weight labeled odd cycle in the graph $G_{\mathcal{S}}$.*

Next we prove one more structural property of labeled intersection graph $G_{\mathcal{S}}$ that will be useful later. We define a (labeled) *spanning tree* T of a connected labeled multi-graph $G_{\mathcal{S}}$ to be a subgraph of $G_{\mathcal{S}}$ that is a tree and connects all vertices in \mathcal{S} . An edge $e = (u, v) \in G_{\mathcal{S}}$ is a *tree edge* if $(u, v) \in T$, otherwise it is called a *non-tree edge*.

► **Lemma 5.** *Let $G_{\mathcal{S}}$ be a connected labeled intersection graph and T be a spanning tree of $G_{\mathcal{S}}$. If $G_{\mathcal{S}}$ contains an odd labeled cycle, then it also contains an odd labeled cycle with exactly one non-tree edge.*

Proof. Let C be an odd cycle in $G_{\mathcal{S}}$ that contains fewest non-tree edges. If C consists of exactly one non-tree edge, we are done. Otherwise, C contains more than one non-tree edge. Let $e = (u, v) \in C$ be a non-tree edge and $C' \subset C$ be the remainder of C without the edge e . Since C is odd labeled, we must have $\text{lab}(C') \neq \text{lab}(e)$.

Let π_{uv} be the unique path connecting u, v in T . This gives us a path π_{uv} with label $\text{lab}(\pi_{uv})$. Recall that $\text{lab}(C') \neq \text{lab}(e)$. We have two cases. (i) If $\text{lab}(\pi_{uv}) \neq \text{lab}(e)$, then we obtain an odd labeled cycle $\pi_{uv} \oplus e$ that has one non-tree edge, namely e , and we are done. (ii) Otherwise, $\text{lab}(\pi_{uv}) = \text{lab}(e) \neq \text{lab}(C')$. This gives us an odd labeled closed walk $W^* = \pi_{uv} \oplus C'$ which contains one less non-tree edge than C . Let $C^* \subseteq W^*$ be an odd-labeled inclusion minimal closed sub-walk of W^* (one such C^* always exists). Therefore, C^* is an odd-labeled cycle in $G_{\mathcal{S}}$ that has fewer non-tree edges than C . But C was chosen to be an odd labeled cycle with fewest non-tree edges, a contradiction. ◀

The above lemma also gives a simple $O(|\mathcal{S}^2|)$ algorithm to *detect* whether there exists an odd label cycle in $G_{\mathcal{S}}$. Specifically, consider an arbitrary spanning tree T of $G_{\mathcal{S}}$ and for each edge not in T , compare its label with the label of the path connecting its endpoints in T .

► **Lemma 6.** *Given a labeled graph $G_{\mathcal{S}}$, there exists an $O(|\mathcal{S}^2|)$ time algorithm to detect whether $G_{\mathcal{S}}$ contains an odd labeled cycle.*

4 Application to Obstacle-removal

We will show how to cast OBSTACLE-REMOVAL as a Labeled ODD CYCLE TRANSVERSAL problem on the graph $G_{\mathcal{S}}$. Recall that in OBSTACLE-REMOVAL problem, we want to remove a set $\mathcal{S}_d \subseteq \mathcal{S}$ of obstacles from the input so that s and t are connected in $\mathcal{S} \setminus \mathcal{S}_d$. Equivalently, we want to select a subset \mathcal{S}_d of obstacles such that the complement set $\mathcal{S} \setminus \mathcal{S}_d$ does not separate s and t . From Lemma 3, it follows that the obstacles $\mathcal{S} \setminus \mathcal{S}_d$ do not separate s and t if and only if $G_{\mathcal{S}}[\mathcal{S} \setminus \mathcal{S}_d]$ does not contain an odd labeled cycle. This gives us the following important lemma.

► **Lemma 7.** *A set of obstacles $\mathcal{S}_d \subseteq \mathcal{S}$ is a solution to OBSTACLE-REMOVAL if and only if the set of vertices \mathcal{S}_d is a solution to ODD CYCLE TRANSVERSAL of $G_{\mathcal{S}}$.*

This allows us to apply the set of existing results for ODD CYCLE TRANSVERSAL to obstacle removal problems. In particular, this readily gives an improved algorithm for OBSTACLE-REMOVAL when parameterized by the solution size (number of removed obstacles). Let

G_S^+ denote the graph G_S where every edge e with $\text{lab}(e) = 0$ is subdivided. Clearly an odd-labeled cycle in G_S has odd length in G_S^+ and vice versa. Applying the FPT algorithm for ODD CYCLE TRANSVERSAL from [18] on the graph G_S^+ gives us the following result.

► **Theorem 8.** *There exists a $2.3146^k n^{O(1)}$ algorithm for OBSTACLE-REMOVAL parameterized by k , the number of removed obstacles.*

This also immediately gives us an $O(\sqrt{\log \text{OPT}})$ approximation for OBSTACLE-REMOVAL by using the best known $O(\sqrt{\log \text{OPT}})$ -approximation [15] for ODD CYCLE TRANSVERSAL on the graph G_S^+ . Observe that instances of obstacle removal are special cases of odd cycle transversal, specifically where the graph G_S is an intersection graph of obstacles. By applying known results on *small diameter decomposition of region intersection graphs*, Kumar et al. [21] obtained a constant factor approximation for OBSTACLE-REMOVAL. In the next section we present an alternative constant factor approximation algorithm. Although our algorithm follows a similar high level approach of using small diameter decomposition of G_S , we give an alternative proof which significantly simplifies the arguments of [21].

Constant Approximation for Obstacle-removal

Our algorithm is based on formulating and rounding a standard LP for labeled odd cycle transversal on a labeled intersection graph G_S . Let $0 \leq x_i \leq 1$ be an indicator variable that denotes whether obstacle S_i is included to the solution or not. The LP formulation which will be referred as HIT-ODD-CYCLES-LP can be written as follows:

$$\begin{aligned} \min \sum_{S_i \in \mathcal{S}} x_i \quad \text{subject to:} \\ \sum_{S_j \in C} x_j \geq 1 \quad \quad \quad \text{for all odd-labeled cycles } C \in G_S \end{aligned}$$

Although this LP has exponentially many constraints, it can be solved in polynomial time using the ellipsoid method with the polynomial time algorithm for minimum weight odd labeled cycle in G_S (Lemma 4) as separation oracle. The next step is to round the fractional solution $\hat{x} = x_1, x_2, \dots, x_n$ obtained from solving the HIT-ODD-CYCLES-LP. We will need some background on small diameter decomposition of graphs.

Small Diameter Decomposition. Given a graph $G = (V, E)$ and a distance function $d : V \rightarrow \mathbb{R}^+$ associated with each vertex, we can define the distance of each edge as $d(e) = d(v) + d(w)$ for every edge $e = (v, w) \in E$. We can then extend the distance function to any pair of vertices $d(u, v)$ as the shortest path distance between u and v in the edge-weighted graph with distance values of edges as edge weights. We use the following result of Lee [17] for the special case of *region intersection graph* over planar graphs.

► **Lemma 9.** *Let $G = (V, E)$ be a node-weighted intersection graph of connected regions in the plane, then for every $\Delta > 0$ there exists a set $X \subseteq V$ of $|X| = O(1/\Delta) \cdot \sum d(v)$ vertices such that the diameter of $G - X$ is at most Δ in the metric d . Moreover, such a set X can be computed in polynomial time.*

For the sake of convenience, we assume that G_S does not contain an obstacle S_i with a self-loop, because if so, we must always include S_i to the solution. Let G_S^* be the underlying unlabeled graph obtained by removing labels and multi-edges from G_S . Since G_S^* is simply the intersection graph of connected regions in the plane, it is easy to show that G_S^* is a region intersection graph over a planar graph (See also Lemma 4.1 [21] for more details.)

Algorithm: Hit-Odd-Cycles. With small diameter decomposition for G_S^* in place, the rounding algorithm is really simple.

- Assign distance values to vertices of $G_S^* = (S, E)$ as $d(S_i) = x_i$, where x_i is the fractional solution obtained from solving HIT-ODD-CYCLE-LP.
- Apply Lemma 9 on graph G_S^* with diameter $\Delta = 1/2$. Return the set of vertices X obtained from applying the lemma as solution.

It remains to show that the set $X \subseteq S$ returned above indeed hits all the odd labeled cycles in G_S . Define a ball $\mathcal{B}(c, R) = \{v \in V : d(c, v) < R - d(v)/2\}$ with center c , radius R and distance metric d defined before. Intuitively, $\mathcal{B}(c, R)$ consists of the vertices that lie strictly inside the radius R ball drawn with c as center.

► **Lemma 10.** X hits all odd labeled cycles in G_S .

Proof. The proof is by contradiction. Let C be an odd labeled cycle such that $C \cap X = \emptyset$. Then C must be contained in a single connected component κ of $G_S - X$. Let v_1 be an arbitrary vertex of C and consider a ball $B = \mathcal{B}(v_1, 1/2)$ of radius $1/2$ centered at v_1 . We have $\kappa \subseteq B$ due to the choice of diameter Δ . Consider the shortest path tree T of ball B rooted at v_1 using the distance function $d(e)$ in the unlabeled graph G_S^* . For every edge $(u, v) \in T$ assign the label $\text{lab}(e)$ of $e = (u, v) \in G_S$. If multiple labeled edges exist between u and v , choose one arbitrarily.

Now consider the induced subgraph $G'_S = G_S[B]$ which is a connected labeled intersection graph of obstacles in the ball B . Moreover, T is a spanning tree of G'_S , and G'_S contains an odd-labeled cycle because $\kappa \subseteq G'_S$. Applying Lemma 5 gives us an odd-labeled cycle $C \in G'_S$ that contains exactly one edge $e \notin T$. The cost of this cycle is $\text{cost}(C) < 1/2 + 1/2 = 1$. This contradicts the constraint of HIT-ODD-CYCLE-LP corresponding to C . ◀

We conclude with the main result for this section.

► **Theorem 11.** *There exists a polynomial time constant factor approximation algorithm for OBSTACLE-REMOVAL.*

5 Generalized Points-separation

So far, we have focused on separating a pair of points s, t in the plane. In this section, we consider the more general problem where we are given a set S of n obstacles, a set of points A and a set $P = \{(s_1, t_1), \dots, (s_p, t_p)\}$ of p pairs of points in A which we want to separate. First we show how to extend the labeled intersecting graph G_S to p source-destination pairs and that the optimal solution subgraph $G_S[\mathcal{S}_{OPT}]$ exhibits a “nice” structure. Then we exploit this structure to obtain an $2^{O(p^2)}n^{O(p)}$ exact algorithm for GENERALIZED POINTS-SEPARATION. Since $p = O(k^2)$, this algorithm runs in polynomial time for any fixed k , resolving an open question of [5]. Using a more sophisticated approach, we later show how to improve the running time to $2^{O(p)}n^{O(k)}$.

5.1 A $2^{O(p^2)}n^{O(p)}$ Algorithm

Recall the construction of the labeled intersection graph G_S for a single point pair (s, t) from Section 3. The label $\text{lab}(e) \in \{0, 1\}$ of each edge $e \in G_S$ denotes the parity of edge e with respect to reference curve π connecting s and t . As we generalize the graph $G_S = (S, E)$ to p point pairs, we extend the label function $\text{lab} : E \rightarrow \{0, 1\}^p$ as a p -bit binary string that denotes the parity with respect to reference curve π_i connecting s_i and t_i for all $i \in [p]$. We will use $\text{lab}_i(e)$ to denote the i -th bit of $\text{lab}(e)$.

Generalized Label Intersection Graph.

- For each $(s_i, t_i) \in P$ and each $S \in \mathcal{S}$ that contains at least one of s_i or t_i , we add a self loop e on S with $\text{lab}_i(e) = 1$ and $\text{lab}_j(e) = 0$ for all $j \neq i$.
- For every pair of intersecting obstacles S, S' and a p -bit string $\ell \in \{0, 1\}^p$:
 - Let $\Pi = \{\pi_i \mid s_i, t_i \notin S \cup S'\}$ be the set of reference curves that do not have endpoints in $S \cup S'$.
 - We add an edge $e = (S, S')$ with $\text{lab}(e) = \ell$ if there exists a plane curve connecting $\text{ref}(S)$ and $\text{ref}(S')$ contained in $S \cup S'$ that crosses all reference curves $\pi_i \in \Pi$ with parity consistent with label ℓ . That is, the curve crosses π_i and odd (resp. even) number of times if i -th bit of ℓ is 1 (resp. 0).

Similar to the one pair case, we can build an unlabeled graph G' with vertex set $\mathcal{S} \times \{0, 1\}^p$ and edges between them based on the arrangement $\text{Arr}(S \cup S' \cup \pi_1 \cup \dots \cup \pi_p)$. Using this graph, we can obtain the following lemma.

► **Lemma 12.** *The generalized label intersection graph $G_{\mathcal{S}}$ with p -bit labels can be constructed in $2^{O(p)}n^{O(1)}$ time.*

Suppose we define $G_{\mathcal{S}}(i)$ to be the image of $G_{\mathcal{S}}$ induced by the labeling $\text{lab}_i : E \rightarrow \{0, 1\}$. Specifically, we obtain $G_{\mathcal{S}}(i)$ from $G_{\mathcal{S}}$ by replacing label of each edge by the i -th bit $\text{lab}_i(e)$, followed by removing parallel edges that have the same label. Observe that $G_{\mathcal{S}}(i)$ is precisely the graph obtained by applying algorithm from Section 3 with reference curve π_i . We say that a subgraph $G'_{\mathcal{S}} \subseteq G_{\mathcal{S}}$ is *well-behaved* if $G'_{\mathcal{S}}(i)$ contains an odd labeled cycle for all $i \in [p]$. The following lemma can be obtained by applying Lemma 3 for every pair $(s_i, t_i) \in P$.

► **Lemma 13.** *A set of obstacles $\mathcal{S}' \subseteq \mathcal{S}$ separate all point pairs in P iff $G_{\mathcal{S}}[\mathcal{S}']$ is well-behaved.*

We will prove the following important property of well-behaved subgraphs of $G_{\mathcal{S}}$.

► **Lemma 14.** *Let $G \subseteq G_{\mathcal{S}}$ be an inclusion minimal well-behaved subgraph of $G_{\mathcal{S}}$. Then there exists a set $V_c \subseteq V(G)$ of connector vertices such that G consists of the vertex set V_c and a set of K chains (path of degree 2 vertices) with endpoints in V_c . Moreover, $|V_c| \leq 4p$ and $|K| \leq 5p$.*

Proof. Since G is an inclusion minimal well-behaved subgraph, it does not contain a proper subgraph that is also well-behaved. Therefore, G does not contain a vertex of degree at most 1 because such vertices and edges adjacent to them cannot be part of any cycle. Suppose G has r connected components C_1, \dots, C_r . We fix a spanning tree T_j of C_j for each $j \in [r]$. We construct the set V_c by including every vertex of degree three or more to V_c . The components C_j that do not contain a vertex of degree three must be a simple cycle because G does not have degree-1 vertices. For every such C_j , we include vertices adjacent to the only non-tree edge of C_j . It is easy to verify that G consists of K chains connecting vertices in V_c .

Let E_0 be the set of non-tree edges, that are edges not in T_j for some $j \in [r]$. We claim that $|E_0| \leq p$. Since G is well-behaved, $G(i)$ consists an odd-labeled cycle for all $i \in [p]$. Using Lemma 5, and the spanning tree T_j of the component containing that odd labeled cycle, we can transform into an odd-labeled cycle that uses at most one non-tree edge. Repeating this for all pairs, we can use at most p edges from E_0 . If $|E_0| > p$, then we would have a proper subgraph of G with at most p edges that is also well-behaved, which is not possible because G was chosen to be inclusion minimal. Therefore $|E_0| \leq p$.

The graph G only contains vertices of degree 2 or higher, hence each leaf node of the trees T_1, \dots, T_r must be adjacent to some edge in E_0 . Therefore, the number of leaf nodes is at most $2p$, and so the number of nodes of degree three or above in T_1, \dots, T_r is also at most

$2p$. Observe that the vertices in V_c are either adjacent to some edge in E_0 or have degree three or more in some tree T_j . The number of both these type of vertices is at most $2p$, which gives us $|V_c| \leq 4p$. Finally, we bound $|K|$, the number of chains. Note that each edge of G belongs to exactly one chain in K . Therefore, the number of chains containing at least one edge in E_0 is at most p , because $|E_0| \leq p$. All the other chains that do not have any edge in E_0 , are contained in the trees T_1, \dots, T_r . It follows that these chains do not form any cycle, and thus their number is less than $|V_c|$. This gives us $|K| \leq 5p$. ◀

It is easy to see that if $\mathcal{S}' \subseteq \mathcal{S}$ is an optimal set of obstacles separating all pairs in P , then there exists an inclusion minimal well-behaved subgraph G of $G_{\mathcal{S}}[\mathcal{S}']$ that satisfies the property of Lemma 14. Observe that the K chains of graph G are vertex disjoint, so for every chain K_t connecting vertices $S_i, S_j \in V_c$ that has $\text{lab}(K_t) = \ell$, an optimal solution will always choose the walk in $G_{\mathcal{S}}$ that has label ℓ and has fewest vertices. To that end, we will need the following simple lemma which is a generalization of the algorithm to compute shortest odd cycle in $G_{\mathcal{S}}$ with 1-bit labels.

► **Lemma 15.** *Given a labeled graph $G_{\mathcal{S}} = (\mathcal{S}, E)$ with labeling $\text{lab} : E \rightarrow \{0, 1\}^p$, the shortest walk between any pair of vertices S_i, S_j with a fixed label $\ell \in \{0, 1\}^p$ can be computed in $2^{O(p)}n^{O(1)}$ time.*

Algorithm: Separate-Point-Pairs.

1. For every pair of vertices $S_i, S_j \in \mathcal{S}$ and every label $\ell \in \{0, 1\}^p$, precompute the shortest walk connecting S_i, S_j with label ℓ in $G_{\mathcal{S}}$ using Lemma 15.
2. For all possible sets $V_c \subseteq \mathcal{S}$ and ways of connecting V_c by K chains:
 - For all $(2^p)^{5p} = 2^{O(p^2)}$ possible labeling of K chains:
 - a. Let $G \subseteq G_{\mathcal{S}}$ be the labeled graph consisting of vertices V_c and chains $K_t \in K$ replaced by shortest walk between endpoints of K_t with label $\text{lab}(K_t)$, already computed in Step 1.
 - b. Check if the graph G is well-behaved. If so, add its vertices as one candidate solution.
3. Return the candidate vertex set with smallest size as solution.

Precomputing labeled shortest walks in Step 1 takes at most $2^{O(p)}n^{O(p)}$ time. The total number of candidate graphs G is $n^{O(p)} \cdot p^{O(p)} \cdot 2^{O(p^2)}$, and checking if it is well behaved can be done in $n^{O(1)}$ time. We have the following result.

► **Theorem 16.** *GENERALIZED POINTS-SEPARATION for connected obstacles in the plane can be solved in $2^{O(p^2)}n^{O(p)}$ time, where n is the number of obstacle and p is the number of point-pairs to be separated.*

► **Corollary 17.** *POINT-SEPARATION for connected obstacles in the plane can be solved in $2^{O(k^4)}n^{O(k^2)}$ time, where n is the number of obstacles and k is the number of points. This is polynomial in n for every fixed k .*

5.2 Faster Algorithms for Points-separation

Recall that the labeled graph $G_{\mathcal{S}}$ constructed in the previous section consisted of labels that are p -bit binary strings. As a result, the running time has a dependence of $n^{O(p)}$ which in worst case could be $n^{O(k^2)}$, for example, in the case of POINTS-SEPARATION when P consists of all point pairs. In this section, we describe an alternative approach that builds a labeled intersection graph whose labels are k -bit strings. Using this graph and the notion of *parity*

partitions, we obtain an $2^{O(p)}n^{O(k)}$ algorithm for GENERALIZED POINTS-SEPARATION which gets rid of the $n^{O(k^2)}$ dependence for POINTS-SEPARATION. Due to lack of space, we describe our approach at a high level and defer the details to the full paper.

The construction of graph G_S is almost the same as before, except that now we choose the reference curves π_i differently. In particular, let $A = \{a_1, a_2, \dots, a_k\}$ be the set of points and P be a set of pairs (a_i, a_j) of points we want to separate. We pick an arbitrary point o in the plane, and for each $i \in [k]$, we fix a plane curve with endpoints a_i and o as the reference curve π_i . For an edge e , the parity of crossing with respect to π_i defines the i -th bit of $\text{lab}(e)$. The graph G_S constructed in this fashion has k -bit labels and will be referred as k -labeled graph.

Let G be a k -labeled graph. For a cycle (or a path) γ in G with edge sequence (e_1, \dots, e_r) , we define $\text{parity}(\gamma) = \bigoplus_{t=1}^r \text{lab}(e_t)$ and denote by $\text{parity}_i(\gamma)$ the i -th bit of $\text{parity}(\gamma)$ for $i \in [k]$. Here the notation “ \oplus ” denotes the bitwise XOR operation for binary strings. Also, we define $\Phi(\gamma)$ as the partition of $[k]$ consisting of two parts $I_0 = \{i : \text{parity}_i(\gamma) = 0\}$ and $I_1 = \{i : \text{parity}_i(\gamma) = 1\}$. Next, we define an important notion called *parity partition*.

► **Definition 18** (parity partition). *Let G be a k -labeled graph. The **parity partition** induced by G , denoted by Φ_G , is the partition of $[k]$ such that $i, j \in [k]$ belong to the same part of Φ_G iff $\text{parity}_i(\gamma) = \text{parity}_j(\gamma)$ for every cycle γ in G .*

We say a k -labeled graph G is P -good if for all $(i, j) \in P$, i and j belong to different parts in Φ_G . The notion of P -goodness in k -labeled graphs is similar to *well-behaved* property of subgraphs G'_S that we defined in Lemma 13 except that the latter is defined using p reference curves. We prove the following lemma that establishes a characterization of obstacles that separate all point pairs in P called P -separators using P -goodness.

► **Lemma 19.** *A subset $S' \subseteq S$ is a P -separator iff the induced subgraph $G_S[S']$ is P -good.*

Similar to Lemma 14, one can show that there exists a P -good subgraph with $4k$ vertices and $5k$ edges. Applying the algorithm SEPARATE-POINT-PAIRS from previous section gives an improved bound of $2^{k^2}n^{O(k)}$. Improving the running time to $2^{O(p)}n^{O(k)}$ require further nontrivial efforts. We defer the details to full version and state our main results.

► **Theorem 20.** *GENERALIZED POINT-SEPARATION for connected obstacles in the plane can be solved in $2^{O(p)}n^{O(k)}$ time, where n is the number of obstacles, k is the number of points, and p is the number of point-pairs to be separated.*

► **Corollary 21.** *POINT-SEPARATION for connected obstacles in the plane can be solved in $2^{O(k^2)}n^{O(k)}$ time, where n is the number of obstacles and k is the number of points.*

Even Faster Algorithm for Pseudo-Disk Obstacles. If the obstacles in S are pseudo-disks then we can further improve the dependence on n to be $n^{O(\sqrt{k})}$. To this end, the key observation is the following analog of Lemma 19 for pseudo-disk obstacles.

► **Lemma 22.** *Suppose S consists of pseudo-disk obstacles. Then a subset $S' \subseteq S$ is a P -separator iff there is a subgraph of the induced subgraph $G_S[S']$ that is planar and P -good.*

The planarity of subgraph $G_S[S']$ allows us to efficiently enumerate the candidate sets using the *planar separator theorem*. We state our main result for such obstacles.

► **Theorem 23.** *GENERALIZED POINT-SEPARATION for pseudo-disk obstacles in the plane can be solved in $2^{O(p)}k^{O(k)}n^{O(\sqrt{k})}$ time, where n is the number of obstacles, k is the number of points, and p is the number of point-pairs to be separated.*

► **Corollary 24.** POINT-SEPARATION for pseudo-disk obstacles in the plane can be solved in $2^{O(k^2)}n^{O(\sqrt{k})}$ time, where n is the number of obstacles and k is the number of points.

5.3 Hardness of Points-separation

We complement our algorithmic results for POINTS-SEPARATION with almost matching hardness bounds assuming the Exponential Time Hypothesis (ETH). We obtain the following results by reductions from PARTITIONED SUBGRAPH ISOMORPHISM [19] and PLANAR MULTIWAY CUT [20] respectively.

► **Theorem 25.** Unless ETH fails, a POINTS-SEPARATION instance (\mathcal{S}, A) for general obstacles cannot be solved in $f(k)n^{o(k/\log k)}$ time where $n = |\mathcal{S}|$ and $k = |A|$.

► **Theorem 26.** Unless ETH fails, a POINTS-SEPARATION instance (\mathcal{S}, A) with pseudodisk obstacles cannot be solved in $f(k)n^{o(\sqrt{k})}$ time where $n = |\mathcal{S}|$ and $k = |A|$.

References

- 1 Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $O(\sqrt{\log n})$ approximation algorithms for min uncut, min 2cnf deletion, and directed cut problems. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 573–581, 2005.
- 2 Paul Balister, Zizhan Zheng, Santosh Kumar, and Prasun Sinha. Trap coverage: Allowing coverage holes of bounded diameter in wireless sensor networks. In *IEEE INFOCOM 2009*, pages 136–144. IEEE, 2009.
- 3 Sayan Bandyopadhyay, Neeraj Kumar, Subhash Suri, and Kasturi Varadarajan. Improved approximation bounds for the minimum constraint removal problem. *Computational Geometry*, 90:101650, 2020.
- 4 Sergey Bereg and David G. Kirkpatrick. Approximating barrier resilience in wireless sensor networks. In *Proc. of 5th ALGOSENSORS*, volume 5804, pages 29–40, 2009.
- 5 S. Cabello and P. Giannopoulos. The complexity of separating points in the plane. *Algorithmica*, 74(2):643–663, 2016.
- 6 David Yu Cheng Chan and David G. Kirkpatrick. Approximating barrier resilience for arrangements of non-identical disk sensors. In *Proc. of 8th ALGOSENSORS*, pages 42–53, 2012.
- 7 David Yu Cheng Chan and David G. Kirkpatrick. Multi-path algorithms for minimum-colour path problems with applications to approximating barrier resilience. *Theor. Comput. Sci.*, 553:74–90, 2014.
- 8 E. Eiben and I. Kanj. How to navigate through obstacles? In *Proc. of 45th ICALP*, 2018.
- 9 Eduard Eiben, Jonathan Gemmell, Iyad A. Kanj, and Andrew Youngdahl. Improved results for minimum constraint removal. In *Proc. of 32nd AAAI*, pages 6477–6484, 2018.
- 10 Eduard Eiben and Iyad Kanj. A colored path problem and its applications. *ACM Trans. Algorithms*, 16(4):47:1–47:48, 2020.
- 11 Eduard Eiben and Daniel Lokshtanov. Removing connected obstacles in the plane is FPT. In *Proc. of 36th SoCG*, volume 164, pages 39:1–39:14, 2020.
- 12 Lawrence H. Erickson and Steven M. LaValle. A simple, but NP-Hard, motion planning problem. In *Proc. of 27th AAAI*, 2013.
- 13 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
- 14 Matias Korman, Maarten Löffler, Rodrigo I. Silveira, and Darren Strash. On the complexity of barrier resilience for fat regions and bounded ply. *Comput. Geom.*, 72:34–51, 2018.
- 15 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *Journal of the ACM (JACM)*, 67(3):1–50, 2020.

52:14 Algorithms for Point Separation and Obstacle Removal

- 16 Santosh Kumar, Ten-Hwang Lai, and Anish Arora. Barrier coverage with wireless sensors. *Wirel. Networks*, 13(6):817–834, 2007.
- 17 James R. Lee. Separators in region intersection graphs. In *Proc. of 8th ITCS*, volume 67, pages 1–8, 2017.
- 18 Daniel Lokshantov, NS Narayanaswamy, Venkatesh Raman, MS Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Transactions on Algorithms (TALG)*, 11(2):1–31, 2014.
- 19 Dániel Marx. Can you beat treewidth? In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 169–179. IEEE, 2007.
- 20 Dániel Marx. A tight lower bound for planar multiway cut with fixed number of terminals. In *International Colloquium on Automata, Languages, and Programming*, pages 677–688. Springer, 2012.
- 21 Saket Saurabh Neeraj Kumar, Daniel Lokshantov and Subhash Suri. A constant factor approximation for navigating through connected obstacles in the plane. In *Proc. 32nd SODA*, 2021.

A Universal Triangulation for Flat Tori

Francis Lazarus  

G-SCOP, CNRS, UGA, Grenoble, France

Florent Tellerie 

G-SCOP, UGA, Grenoble, France

Abstract

A result due to Burago and Zalgaller states that every orientable polyhedral surface, one that is obtained by gluing Euclidean polygons, has an isometric piecewise linear (PL) embedding into Euclidean space \mathbb{E}^3 . A flat torus, resulting from the identification of the opposite sides of a Euclidean parallelogram, is a simple example of polyhedral surface. In a first part, we adapt the proof of Burago and Zalgaller, which is partially constructive, to produce PL isometric embeddings of flat tori. In practice, the resulting embeddings have a huge number of vertices, moreover distinct for every flat torus. In a second part, based on another construction of Zalgaller and on recent works by Arnoux et al., we exhibit a *universal triangulation* with 5974 triangles which can be embedded linearly on each triangle in order to realize the metric of any flat torus.

2012 ACM Subject Classification Mathematics of computing \rightarrow Geometric topology; Mathematics of computing \rightarrow Discrete mathematics; Theory of computation \rightarrow Computational geometry

Keywords and phrases Triangulation, flat torus, isometric embedding

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.53

Related Version *Full Version:* <https://arxiv.org/abs/2203.05496>

Funding *Francis Lazarus:* This author is partially supported by the French ANR projects GATO (ANR-16-CE40-0009-01) and MINMAX (ANR-19-CE40-0014) and the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program Investissement d’avenir.

Acknowledgements We warmly thank Alba Málaga, Pierre Arnoux and Samuel Lelièvre for sharing with us their constructions of flat tori and showing us how to cover their moduli space with these constructions. We are also grateful to the anonymous reviewers for their careful reading and suggestions.

1 Introduction

A celebrated theorem of Nash [6] completed by Kuiper [5] implies that every smooth Riemannian orientable surface has a C^1 isometric embedding in the Euclidean 3-space \mathbb{E}^3 . As a consequence one can represent and visualize faithfully in \mathbb{E}^3 the geometry of any abstract orientable Riemannian surface. An analogous result, due to Burago and Zalgaller [3], states that every orientable polyhedral surface, obtained by abstractly gluing Euclidean polygons, has an isometric piecewise linear (PL) embedding in \mathbb{E}^3 . In particular, this provides PL isometric embeddings for every flat torus, the result of the identification of the opposite sides of a Euclidean parallelogram. However, the proof of Burago and Zalgaller is partially constructive, relying on the subdivision of the polyhedral surface into an acute triangulation and on the Nash-Kuiper theorem itself, which is a priori far from constructive. The singular vertices of the polyhedral surface (where the angles at the incident polygons do not sum up to 2π) moreover deserve special treatments with several constants that are rather hard to estimate. In the case of flat tori, all these difficulties can be circumvented. In particular, a flat torus has no singular vertex. Using a simple construction of acute triangulations together with the conformal embeddings of Hopf-Pinkall [7, 2], we were able to compute PL isometric embeddings of various flat tori, including the square and the hexagonal tori.



© Francis Lazarus and Florent Tellerie;

licensed under Creative Commons License CC-BY 4.0

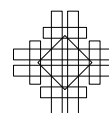
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 53; pp. 53:1–53:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



In practice, the construction of Burago and Zalgaller, even including our simplifications for flat tori, produces PL embeddings with a huge number of vertices: more than 170,000 for the square torus and more than 7 millions for the hexagonal torus. Most importantly, the underlying triangulations of the resulting PL embeddings depends on the geometry (or *modulus*) of the flat tori and are pairwise non-isomorphic. Apart from the construction of Burago and Zalgaller, describing explicit PL embeddings of specific flat tori does not seem a simple task. As an illustrating example, it was only very recently that an explicit PL embedding of the square flat torus appeared in the literature [8].

We say that a triangulation of the topological torus is *universal* if, for any flat torus, it admits a geometric realization in \mathbb{E}^3 that is isometric to this flat torus. It is not clear that such a universal triangulation should exist as the moduli space of flat tori is not compact. In particular, there is no reason why any of the triangulations obtained from the method of Burago and Zalgaller would be universal. Our main result is the rather surprising existence of a universal triangulation with the description of such a triangulation of reasonable size.

► **Theorem 1.** *There exists an abstract triangulation \mathcal{T} of the torus with 5974 triangles that admits for each flat torus (in the moduli space) an embedding in \mathbb{E}^3 which is linear on each triangle of \mathcal{T} , and which is isometric to this flat torus.*

2 Background and definitions

Polyhedral surfaces

A **polyhedral surface** is a compact topological surface obtained from a finite collection of polygonal regions in the Euclidean plane by gluing their sides according to a partial oriented pairing. This pairing should be such that each side is paired at most once and two sides in a pair should have the same length. The pair orientation specifies one of the two isometries between its sides. Since every polygon can be triangulated, one can replace the polygons by triangles in this definition. The collection of triangles together with their gluing determine a **triangulation** of the surface. This triangulation is **simplicial** when there is no loop edge or parallel edges. By an **abstract triangulation** of a polyhedral surface, we mean a simplicial complex that is isomorphic to some triangulation of the polyhedral surface.

Polyhedral metric

The gluing of Euclidean polygons induces a **length metric** on the resulting polyhedral surface: the distance between any two points is the infimum of the lengths of the paths connecting the two points. There is an intrinsic definition of polyhedral surfaces that does not assume any specific decomposition into polygons. A **polyhedral metric** on a topological surface is a metric such that every point has a neighborhood isometric to a neighborhood of the apex of a Euclidean cone. In turn, a (2-dimensional) Euclidean cone is defined by coning from the origin a rectifiable simple curve on the unit sphere in \mathbb{E}^3 . The length of this curve is the total angle of the cone. A point whose conic neighborhood has total angle different from 2π is called a **singular vertex**.

Piecewise linear maps and isometries

Let S be a polyhedral surface. A map $f : S \rightarrow \mathbb{E}^3$ is said **piecewise linear** (PL) if S admits a triangulation such that the restriction of f to any triangle is *linear*, *i.e.*, it preserves barycentric coordinates. Once a triangulation of S is given, the image of its vertices in \mathbb{E}^3 determines a unique **linear map** on this triangulation by extending linearly to the images of triangles.

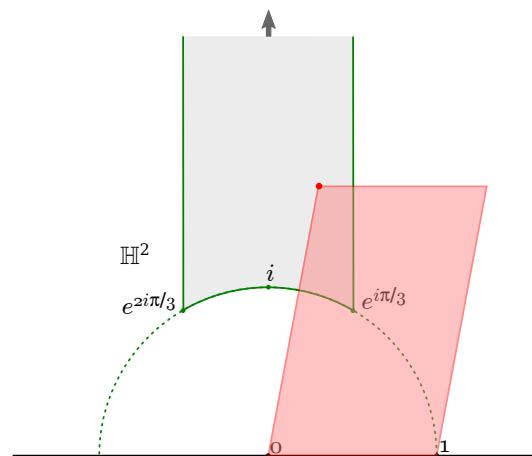
f is **piecewise distance preserving** if S admits a triangulation such that the restriction of f to any triangle is distance preserving, *i.e.*, $|f(x) - f(y)| = d_S(x, y)$ for any x, y in a same triangle. Here, $|\cdot|$ is the Euclidean norm and d_S is the polyhedral metric on S . In particular, the piecewise distance preserving map f must be **length preserving**: if $\gamma : [a, b] \rightarrow S$ is a rectifiable path, then γ and its image $f \circ \gamma$ have the same length. The map f is an **embedding** if it induces a homeomorphism onto its image $f(S)$ endowed with the restriction of the topology of \mathbb{E}^3 . In that case, $f(S)$ is naturally equipped with a length metric induced by the Euclidean metric of \mathbb{E}^3 so that the length of a path in $f(S)$ is its Euclidean length as a path in \mathbb{E}^3 .

A length preserving embedding is the same as an **isometry** between S and $f(S)$, where each surface is endowed with its own length metric, respectively polyhedral and induced by the Euclidean metric. Thus, a piecewise distance preserving embedding is the same as a **PL isometric embedding**. A map $f : S \rightarrow \mathbb{E}^3$ is said **contracting**, or **short**, if there is a constant $C < 1$ such that $|f(x) - f(y)| \leq C d_S(x, y)$ for all $x, y \in S$.

Flat tori

A **flat torus** is a polyhedral surface obtained from a Euclidean parallelogram by pairing its opposite sides. We usually consider flat tori up to re-scaling. This amounts to consider that similar parallelograms lead to the same flat torus. If (e_1, e_2) is the canonical basis of the Euclidean plane, we can thus assume that the two sides of the parallelogram are respectively e_1 and τ for some vector $\tau = \tau_1 e_1 + \tau_2 e_2$, with $\tau_i > 0$. Identifying the real plane with the complex line, we conclude that a flat torus is determined by its **modulus** $\tau = \tau_1 + i\tau_2$.

Rather than gluing the sides of a parallelogram, one can equivalently obtained the same flat torus by quotienting the Euclidean plane \mathbb{E}^2 by the rank 2 lattice $\mathbb{Z}\tau + \mathbb{Z}e_1$ acting by translations. The same lattice is generated by the vectors $(a\tau + b, c\tau + d)$, where $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \text{SL}_2(\mathbb{Z})$ is an integer matrix with determinant 1. This lattice corresponds to the modulus $(a\tau + b)/(c\tau + d)$, where τ is again viewed as a complex number. In fact, the set of flat tori is in one-to-one correspondence with the quotient $\mathbb{H}^2/\text{SL}_2(\mathbb{Z})$, where \mathbb{H}^2 denotes the upper half-plane (the set of moduli) and $\text{SL}_2(\mathbb{Z})$ acts as above. Every flat torus has a modulus in the fundamental domain of this quotient as shown in Figure 1.



■ **Figure 1** A point (in red) in a fundamental domain of the moduli space of tori (in light grey) with the corresponding parallelogram.

3 The construction of Burago and Zalgaller

We first recall the result of Burago and Zalgaller for embedded surfaces.

► **Theorem 2** (Burago and Zalgaller [3]). *Every short C^2 embedding in \mathbb{E}^3 of a polyhedral surface can be approximated by a PL isometric embedding.*

Here, the approximation by a PL isometric map means that for any $\varepsilon > 0$ there is such a map moving the points of the short C^2 -embedding by a distance less than ε . This implies that every orientable polyhedral surface has an isometric PL embedding in 3-space. Before we give a sketch of the proof, we describe the basic construction of Burago and Zalgaller, which is a specialization of Theorem 2 to the case of a single triangle.

3.1 Embedding a triangle

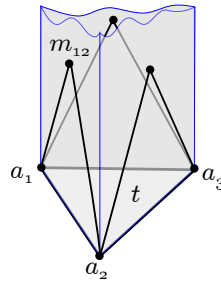
If t is a triangle in \mathbb{E}^3 and \vec{n} is a vector normal to t , then the **prism above** t is the set $\{p + \lambda\vec{n} \mid p \in t, \lambda \geq 0\}$ and the three infinite faces of this prism are its **walls**.

► **Lemma 3** ([3]). *Let $T = A_1A_2A_3$ and $t = a_1a_2a_3$ be (Euclidean) triangles in \mathbb{E}^3 such that*

- (i) *T and t are acute,*
- (ii) *$|a_ia_j| < |A_iA_j|$ for $i, j = 1, 2, 3; i \neq j$,*
- (iii) *the distance of the circumcenter ω of t to each side a_ia_j is smaller than the distance of the circumcenter Ω of T to the corresponding side A_iA_j .*

Denote by m_{ij} the point in the wall above a_ia_j at equal distance from a_i and a_j . Then, T has a PL isometric embedding in the prism above t (with respect to a normal directions) with the boundary condition that each side A_iA_j is sent to the broken line $a_ism_{ij}a_j$.

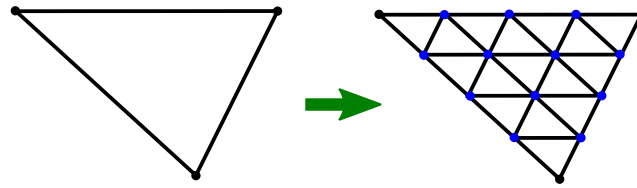
This Lemma (see Figure 2) easily implies that T has a PL isometric embedding arbitrarily



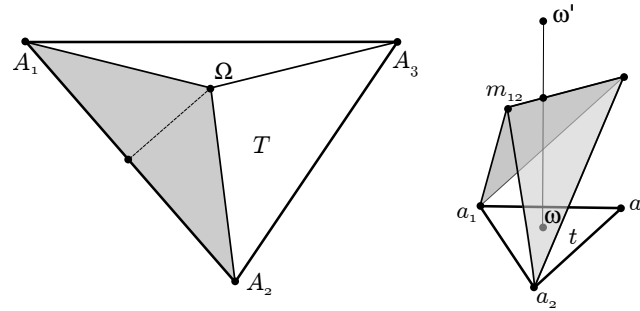
■ **Figure 2** The prism above t .

close to t . Indeed, by subdividing T and t uniformly as in Figure 3 we get similar triangles of smaller size to which we can individually apply Lemma 3. Thanks to the boundary condition in the lemma, the individual constructions fit together to form an isometric embedding of T . The constructions for the smaller triangles being homothetic to the construction for the original triangles, we get closer and closer to t as we refine the uniform subdivisions.

The triangles T and t being acute, they contain their circumcenters Ω and ω in their interior. Let \vec{n} be a unit vector normal to t and let ω' be the point vertically above ω such that $|a_1\omega'| = |A_1\Omega|$. Refer to Figure 4. Note that ω' is well-defined since by the assumptions (ii) and (iii) the circumradius $|A_1\Omega|$ of T is larger than the circumradius $|a_1\omega|$ of t . For completeness, we recall the proof of Lemma 3. Triangle T is first subdivided into three subtriangles $\Omega A_i A_j$. The goal is to fold each $\Omega A_i A_j$ above $\omega a_i a_j$ with the boundary condition

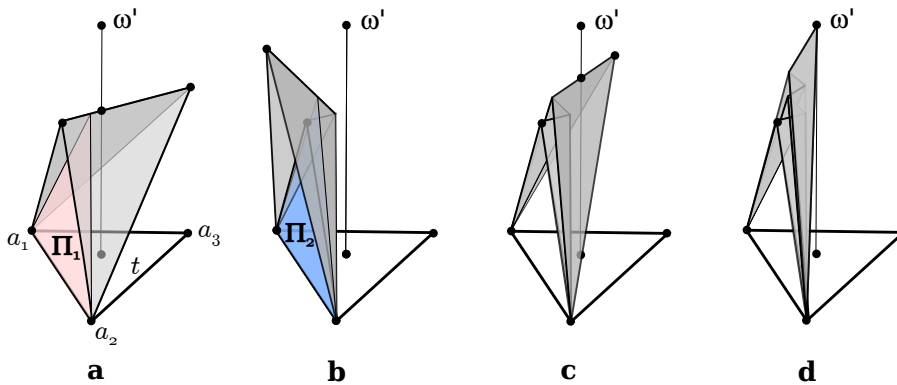


■ **Figure 3** Uniform subdivision of a triangle. The vertices of the subdivision have barycentric coordinates $(i/n, j/n, k/n)$ for $i, j, k \in \mathbb{N}$ and $i + j + k = n$ for some fixed n .



■ **Figure 4** The subtriangle $\Omega A_1 A_2$ is folded above t .

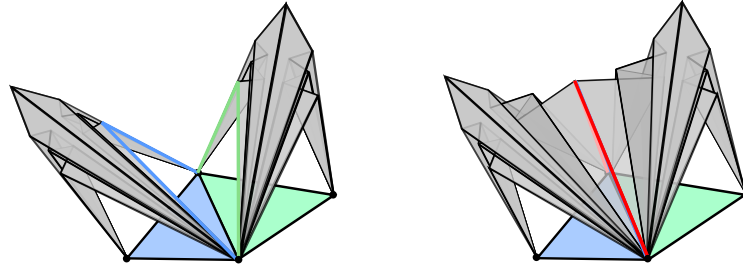
for $A_i A_j$ as in the lemma and so that the boundary edges $\Omega A_i, \Omega A_j$ are sent respectively to the segments $\omega' a_i$ and $\omega' a_j$. To this end, we first fold $\Omega A_1 A_2$ along its altitude from Ω and place the resulting two-winged shape above t so that the side $A_1 A_2$ is folded onto the broken line $a_1 m_{12} a_2$. We next consider a plane Π_1 in the pencil generated by $a_1 a_2$ to reflect the part of the two-winged shape lying to the right of that plane. See Figure 5. Another plane Π_2 in



■ **Figure 5** a, the reflection plane Π_1 . b, after reflection in Π_1 , and the plane Π_2 . c, reflection in Π_2 . d, after an even number of reflections the point Ω is sent to ω' .

the same pencil is then chosen to reflect part of the already reflected part. Choosing Π_1 and Π_2 appropriately, it is not hard to see that after an even number of such reflections the point Ω in $\Omega A_1 A_2$ will be sent to ω' . We finally apply the same construction to the two other subtriangles $\Omega A_2 A_3$ and $\Omega A_3 A_1$ and paste them to form a folding of T above t as desired.

► **Note 4.** This folding of T admits some flexibility. In particular, the boundary conditions can be modified so that each boundary wedge $a_i m_{ij} a_j$ is tilted around the axis $a_i a_j$. This allows to paste the constructions for two adjacent and non coplanar triangles; see Figure 6.



■ **Figure 6** Pasting two foldings of large triangles sharing an edge above smaller triangles that are non coplanar.

3.2 Embedding arbitrary polyhedral surfaces

Denote by $f : S \rightarrow \mathbb{E}^3$ the short C^2 map in Theorem 2. Let U be a union of small polygonal disks centered at each singular vertex of S . The strategy for the proof of Burago and Zalgaller is the following.

- (a) Compute an acute triangulation of $S \setminus U$, where each triangle is acute.
- (b) Compute an approximation f_1 of f that is almost conformal on $S \setminus U$ and short over S .
- (c) Refine the acute triangulation of $S \setminus U$ uniformly to obtain an acute triangulation \mathcal{T} with small triangles. The meaning of *small* depends on the geometric properties of f_1 and on the flexibility in Note 4.
- (d) Replace f_1 by its PL approximation F mapping linearly each triangle $T = A_1A_2A_3$ of \mathcal{T} to the triangle $F(T) := f_1(A_1)f_1(A_2)f_1(A_3)$ in \mathbb{E}^3 .
- (e) Apply the construction in Section 3.1 to every pair $(T, F(T))$, using the tilted version in Note 4 in order to paste the constructions of adjacent triangles.
- (f) Fill the gaps corresponding to U with specific constructions to deal with singularities as described in [3].

We refer to the full version on ArXiv and to the original paper [3] for more details.

4 Embedding flat tori

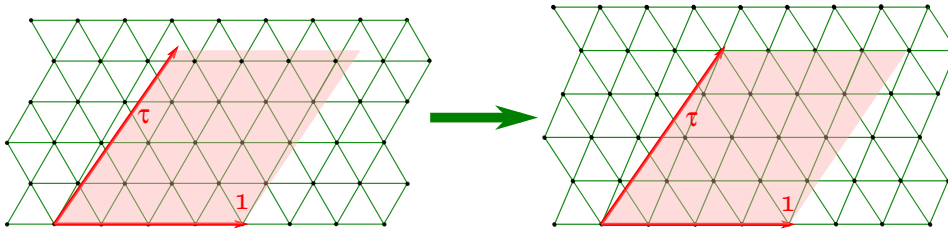
Step (b) in the proof of Burago and Zalgaller is highly non constructive, and to our knowledge no explicit PL isometric embedding of a closed surface according to their method was known up to date. It appears that the steps of their construction can be greatly simplified in the case of flat tori. Thanks to these simplifications we were able to visualize PL isometric embeddings of various flat tori in \mathbb{E}^3 .

We first observe that there is no need for Step (f) since a flat torus has no singular vertex: the angles at the four corners of its defining parallelogram add up to 2π , showing that the only vertex after the side gluing is non singular. In particular, one should set $U = \emptyset$ in all the steps.

4.1 Acute triangulation of flat tori

Itoha and Yuan [4] have shown that every flat torus can be triangulated into at most 16 acute triangles. However, since we need a fine triangulation as in Step (c) with a good control on the acuteness, we use the following triangulation, which is conceptually simpler. Let τ be the modulus of the flat torus $\mathbb{T}_\tau := \mathbb{E}^2 / (\mathbb{Z}\tau + \mathbb{Z}e_1)$ (we abusively identify the plane with the complex numbers). We consider the equilateral triangular lattice generated by $e^{i\pi/3}/n$ and $1/n$ for some positive integer n . This lattice comes with a regular triangulation \mathcal{T}_e by equilateral triangles. Let $p_{a,b} = ae^{i\pi/3}/n + b/n$, with $a, b \in \mathbb{Z}$, be a point in this lattice that is closest to τ . In particular, $|\tau - p_{a,b}| \leq (n\sqrt{3})^{-1}$. We deform \mathcal{T}_e by a linear transformation

ℓ defined by $1 \mapsto 1$ and $p_{a,b} \mapsto \tau$. By the previous inequality and for n large enough, ℓ is close to the identity. The triangles in $\ell(\mathcal{T}_e)$ are thus close to equilateral. Now, the lattice $\mathbb{Z}\tau + \mathbb{Z}e_1$ leaves $\ell(\mathcal{T}_e)$ invariant, so that $\ell(\mathcal{T}_e)/(\mathbb{Z}\tau + \mathbb{Z}e_1)$ is a well defined triangulation of \mathbb{T}_τ by almost equilateral triangles. See Figure 7.



■ **Figure 7** The equilateral triangular lattice (here with $n = 4$) is deformed to fit the lattice of \mathbb{T}_τ .

4.2 Conformal embedding of flat tori

Theorem 2 requires an initial short C^2 embedding, further approximated in Step (b) by an almost conformal map. In the case of flat tori we can directly provide a short conformal embedding. We rely on the Hopf tori developed by Pinkall [7]. These are based on the Hopf fibration

$$p : \mathbb{S}^3 \rightarrow \mathbb{S}^2, (x, y, z, t) \mapsto (2xz + 2yt, 2xt - 2yz, x^2 + y^2 - z^2 - t^2),$$

a standard projection of the 3-sphere \mathbb{S}^3 onto the 2-sphere \mathbb{S}^2 whose fibers (the sets $p^{-1}(s)$ for $s \in \mathbb{S}^2$) are circles. Pinkall proves that if γ is a simple closed curve on \mathbb{S}^2 , then $p^{-1}(\gamma)$ is a flat torus isometric to \mathbb{T}_τ with $\tau = (A + iL)/(4\pi)$, where L is the length of γ and A is the oriented area delimited by γ on \mathbb{S}^2 , choosing the side of γ so that $A \in [-2\pi, 2\pi)$. Since this torus lies in $\mathbb{S}^3 \subset \mathbb{E}^4$, it remains to apply a stereographic projection, say from the South pole $(0, 0, 0, -1)$, assuming it does not lie on the torus, to obtain a conformal embedding of \mathbb{T}_τ in \mathbb{E}^3 . In coordinates: $(x, y, z, t) \mapsto (x, y, z)/(t + 1)$.

Banchoff [2] revisited Pinkall’s approach to give explicit parametrizations of the Hopf-Pinkall tori. On \mathbb{S}^2 , Banchoff considers a curve of the form $\gamma_\tau(\theta) = (\sin \phi(\theta)e^{i\theta}, \cos \phi(\theta))$ given in spherical coordinates, where the polar angle $0 < \phi < \pi$ is a smooth function of the azimuthal angle $0 \leq \theta \leq 2\pi$. He next defines $L(\theta) = \int_0^\theta |\gamma'_\tau(t)| dt$ to be the length of the curve portion $\gamma_\tau([0, \theta])$ and $A(\theta) = \int_0^\theta (1 - \cos \phi(t)) dt$ the area on \mathbb{S}^2 swept by the arc of meridian linking the North Pole to the point on γ_τ up to θ . The conformal embedding $f_\tau : \mathbb{T}_\tau \rightarrow \mathbb{E}^3$ is then given by $f_\tau = f \circ g^{-1}$ with

$$f : (\mathbb{R}/2\pi\mathbb{Z})^2 \rightarrow \mathbb{E}^3, (\theta, \psi) \mapsto \left(\sin \frac{\phi(\theta)}{2} e^{i(\theta+\psi)}, \cos \psi \cos \frac{\phi(\theta)}{2} \right) / \left(1 + \sin \psi \cos \frac{\phi(\theta)}{2} \right), \text{ and}$$

$$g : (\mathbb{R}/2\pi\mathbb{Z})^2 \rightarrow \mathbb{T}_{-1/\tau} \sim \mathbb{T}_\tau, (\theta, \psi) \mapsto \left(\frac{L(\theta)}{2}, \frac{A(\theta)}{2} + \psi \right).$$

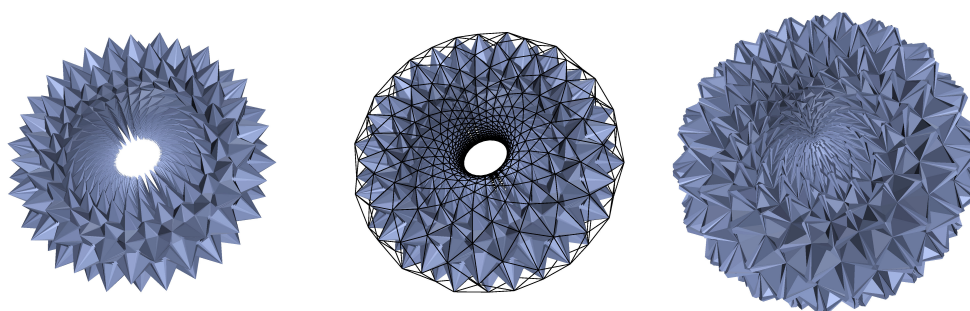
We have chosen ϕ of the form $\phi(\theta) = a + b \sin(n\theta)$ for $a < b, 0 \leq b < \pi - a$ and $n \in \mathbb{N}$. In order to represent the modulus $\tau = \tau_1 + i\tau_i$, the parameters a, b, n should satisfy $A(2\pi) = 4\pi\tau_1$ and $L(2\pi) = 4\pi\tau_i$, or equivalently:

$$J_0(b) \cos(a) = 1 - 2\tau_1 \quad \text{and} \quad \int_0^{2\pi} \sqrt{n^2 b^2 \cos^2(nt) + \sin^2(a + b \sin(nt))} dt = 4\pi\tau_i,$$

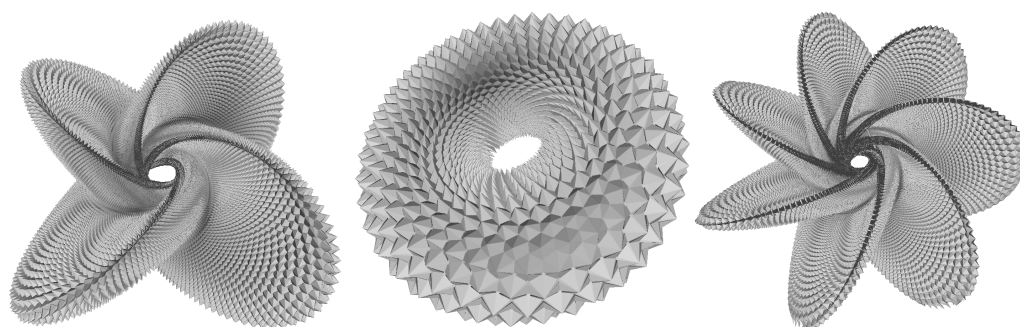
where $J_0(b) = \frac{1}{\pi} \int_0^\pi \cos(b \sin t) dt$ denotes the 0-th Bessel function of the first kind. The condition on the total area implies $0 \leq \tau_1 \leq 1$. Nevertheless, it is still possible to obtain a conformal embedding in the case of $\tau_1 < 0$ by first reflecting the torus along one of its boundary edge and applying a reflexion of the image torus in \mathbb{E}^3 . We can thus cover the whole moduli space.

4.3 The final construction

We now have all the pieces to produce PL isometric embedding of flat tori. Given a modulus τ , we first compute a quasi-equilateral triangulation of \mathbb{T}_τ as in Section 4.1. We then compute a PL approximation F_τ of the conformal map f_τ defined in Section 4.2 and finally apply the construction in Section 3.1 to every pair of triangles $(T, F_\tau(T))$. Figures 8 and 9 show some results.



■ **Figure 8** Left, PL isometric embedding of the square flat torus with 170,040 triangles. Middle, the mesh with black edges shows the PL approximation of the initial almost conformal embedding. Each of its triangles is replaced with a construction (in blue) as in Section 3.1 oriented toward the interior of the initial embedding. Right, The construction is oriented towards the outside, giving another isometric immersion of the square torus. (This last model has self-intersections. A finer triangulation should be used to avoid them.)



■ **Figure 9** isometric immersion of \mathbb{T}_τ with, from left to right, $\tau = e^{i\pi/3}, (1+i)/2, (1+3i)/2$. The left immersion is a hexagonal torus. While the subdivisions of the left and right tori already contain more than 7 millions triangles, they present self-intersections. A finer triangulation should be used to get an embedding.

5 Universal triangulation

The construction of Burago and Zalgaller gives rise to triangulations with a huge number of triangles, moreover distinct for every flat torus. In order to get a unique abstract triangulation that admits linear embeddings in \mathbb{E}^3 isometric to *any* flat torus, we resort to a second construction by Zalgaller [10] and to very recent work by Tsuboi [9] and Arnoux et al. [1] for embedding flat tori.

5.1 Embedding long tori

Any flat torus can be obtained by identifying abstractly the top and bottom boundaries of a right circular cylinder. We obtain non-rectangular tori by shifting circularly the top boundary before identification. We can moreover cover all the torus moduli by varying the ratio between the height of the cylinder and the length of its boundaries. A torus is said **long** when this ratio is large. In [10], Zalgaller proposes an origami style folding of long flat tori, much simpler than the general construction of [3]. Here, we quantify how long should be a torus to allow for the Zalgaller folding, and we show that the long tori admit a universal triangulation.

► **Proposition 5.** *There exists an abstract triangulation with 270 triangles, which admits linear embeddings isometric to every torus of modulus $\tau_1 + i\tau_2$ with $\tau_1 \geq 33$.*

The proof reduces to a careful analysis of the construction of Zalgaller. Instead of a circular cylinder, Zalgaller starts with a polyhedral cylinder in \mathbb{E}^3 , namely a prism with equilateral triangular basis, that he bents at several places to make the boundaries coincide, allowing their geometric identification. A twist is also applied before the bending so as to simulate a circular shift of the top boundary. In general, except for a twist of $2k\pi/3$, one boundary will be rotated with respect to the other after the twisting and bending, preventing their identification. Zalgaller then introduces a third modification that he calls a **gasket** in order to rotate a cross section of the prism without rotating the “material” of the prism. Intuitively, one should imagine a sleeve made of some non-elastic fabric, closed by two rigid triangles at the extremities. The right prism results from pulling tight on the triangles. Now, the effect of a gasket is to rotate one triangle around the axis of the prism, allowing the fabric to *slide* along the edges of this triangle.

How to bend a triangular prism

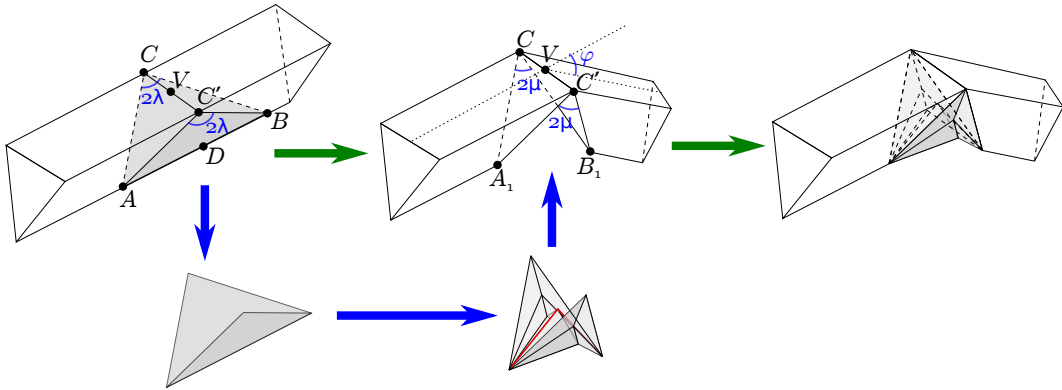
Consider a right prism \mathcal{P} with equilateral triangular basis and an orthogonal cross section $CC'D$. A **bending at an angle φ with cutting angle λ** along the **rib** CC' is obtained by (refer to Figure 10)

- (a) cutting two isosceles triangles ACB and $AC'B$ out of \mathcal{P} , where A, B lie on the generatrix of the prism through D , and the angle at C (and C') is 2λ ,
- (b) bending the cut prism at angle $0 < \varphi < \pi$,
- (c) folding ACB and $AC'B$ appropriately to fit them back on the bended prism.

Let A_1, B_1 be the respective positions of points A, B after bending and let $\angle A_1CB_1 = 2\mu$. In order for the construction not to overlap, one should have $\mu > 0$, hence λ should satisfy $\lambda_0(\varphi) < \lambda < \frac{\pi}{2}$ where $\lambda_0(\varphi)$ is the angle for which, after bending, the triangles A_1CC' and B_1CC' coincide. Looking at the right angled triangles ADC and ADV , one easily computes¹

$$\lambda_0(\varphi) = \arctan\left(\frac{\sqrt{3}}{2} \tan \frac{\varphi}{2}\right). \quad (1)$$

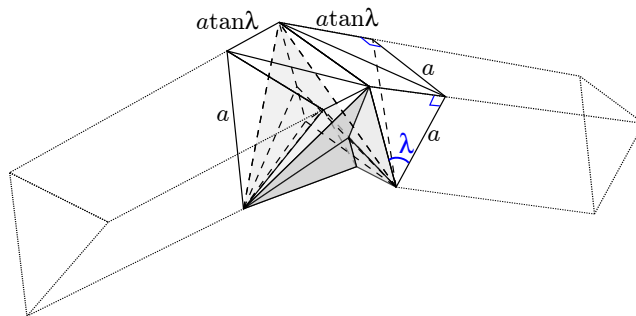
¹ This expression is simpler than the formula given in [10].



■ **Figure 10** Bending of a prism.

► **Lemma 6.** *For every $\varphi \in (0, \pi)$ and for every $\lambda \in (\lambda_0(\varphi), \pi/2)$, there is an embedded bending of \mathcal{P} at angle φ with cutting angle λ introducing 12 triangles.*

See the full version for a proof. For further reference, we call a **bend** a bent prism cut by orthogonal cross sections at the extremity of the above construction as on Figure 11.



■ **Figure 11** A bend is isometric to a right prism of length $2a \tan \lambda$.

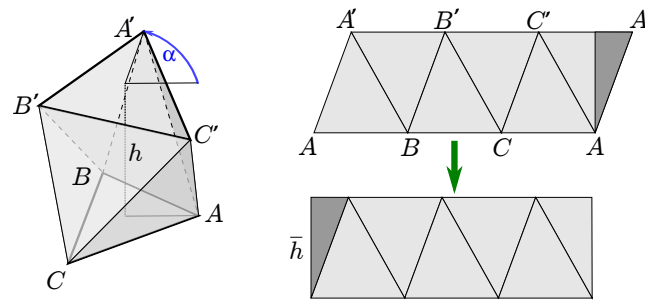
After triangulating the two top quadrilaterals, a bend is made of 20 triangles including the 12 triangles as in Lemma 6.

Rotating a cross section with a gasket

The ribs of the prism \mathcal{P} may have only three possible directions. This prevents to bend \mathcal{P} in an arbitrary direction. To circumvent this rigidity, Zalgaller introduces a simple construction that he calls a gasket. Consider an equilateral triangle ABC in the horizontal plane and a vertical translate $A'B'C'$ by height h . Rotate $A'B'C'$ by an angle α about the central vertical axis. The **gasket with turn α and height h** is the polyhedral cylinder formed by the six congruent triangles ABA' , $A'BB'$, $B'BC'$, $B'CC'$, $C'CA'$, $AA'C'$. See Figure 12. This gasket is embedded for every $\alpha \in (-\pi/3, \pi)$, independently of $h > 0$.

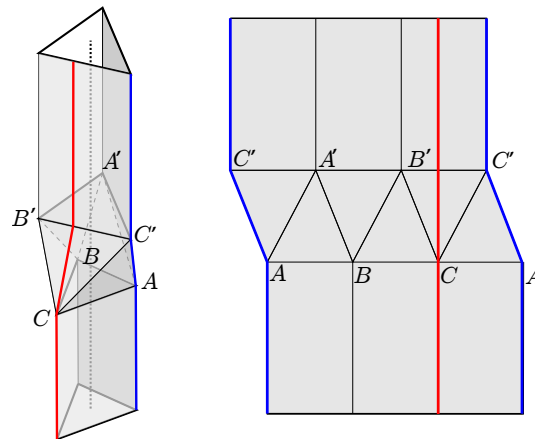
► **Lemma 7.** *For every $\alpha \in (-\pi/3, \pi)$, the gasket with turn α and height h is isometric to a right prism of length \bar{h} with*

$$\bar{h}^2 = h^2 + \frac{2}{27}(\sin^2 \frac{\alpha}{2} + \sin^2(\frac{\pi}{3} - \frac{\alpha}{2})) - \frac{4}{81}(\sin^2 \frac{\alpha}{2} - \sin^2(\frac{\pi}{3} - \frac{\alpha}{2}))^2 - \frac{1}{36} < h^2 + \frac{1}{9}. \quad (2)$$



■ **Figure 12** Left, a gasket with turn α and height h . Right, the gasket is unfolded in the plane. Cutting and pasting a small triangular piece shows that the gasket has the geometry of a right prism.

By pasting two prisms at the boundaries of a gasket, we obtain a polyhedral cylinder with triangular boundaries, where the two boundaries are turned at the angle α with respect to each other; see Figure 13.



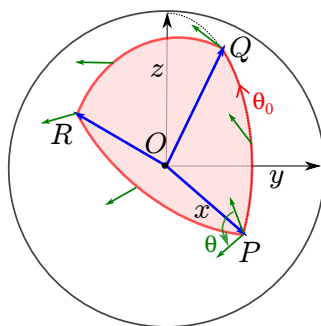
■ **Figure 13** Joining a gasket with two prisms to rotate their ribs. Right, unfolding of the construction showing the line of cut (in blue) and a generatrix (in red) of the polyhedral cylinder.

► **Note 8.** The top and bottom prisms in Figure 13 have the same central axis. This allows to rotate the rib of a prism at an angle $\alpha \in (\pi/3, \pi)$ before applying a bending.

► **Note 9.** By joining k gaskets in a row, we can rotate the rib of a prism at an angle $\alpha \in (-k\pi/3, k\pi)$.

Twisting a prism

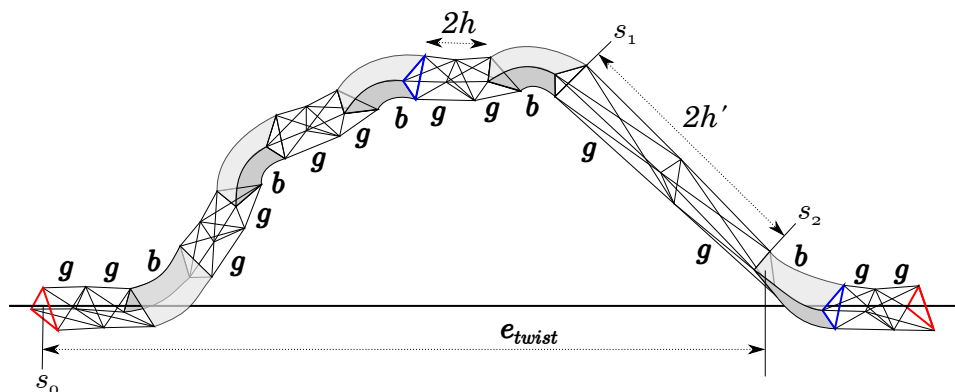
Replacing a portion of a prism by a gasket with turn α allows to turn one boundary, say the top one, of the prism with respect to the other one but does not *twist* the geometry: the top endpoint of a geodesic line perpendicular to the bottom boundary (a generatrix) will glide along the top boundary as we augment α . In order to twist the prism so that the top endpoint of this geodesic line indeed turns with the boundary, Zalgaller introduces yet another construction that he calls a **helical twist**. This construction takes advantage of the holonomy of parallel transport on the sphere: consider a unit sphere of center O with a spherical triangle PQR (see Figure 14). If one parallel transports an object from P to P



■ **Figure 14** The green tangent vector is transported along the spherical triangle PQR . The angle θ is equal to the area of PQR , while the angle θ_0 is given by L'Huillier's formula.

following the sides of the triangle PQR , then the object is rotated by a certain angle around the axis OP which is equal to the signed area of the spherical triangle PQR . In order to twist a prism with axis directed by \vec{OP} by an angle θ we may thus bend the prism successively in the directions \vec{OQ} , \vec{OR} and \vec{OP} again. Each bending at angle φ indeed corresponds to a transport along a spherical geodesic of length φ . Each portion of prism between two bends should include two gaskets to orient its rib properly. Indeed, by Note 9, two gaskets allow to turn by an angle in $(-2\pi/3, 2\pi)$, which covers all the possible orientations.

A **helical twist of angle θ** consists of a sequence of gaskets and bends according to the pattern $(g^2b)^5g^2 = (g^2b)^3(g^2b)^2g^2$, where b, g stand respectively for bends and gaskets. The prefix $(g^2b)^3$ in the pattern is used to simulate the parallel transport as described above, assuming that the central axis of the initial cross section is already aligned with \vec{OP} . The next factor $(g^2b)^2$ allows to return on the central axis of the initial cross section. Since \vec{OP} is aligned with this central axis, the changes of direction due to the factor $(g^2b)^2$ happen in the same plane. The resulting holonomy is thus trivial. Finally, the last two gaskets allows to turn the cross section by any angle in $(-2\pi/3, 2\pi]$; see Figure 15.



■ **Figure 15** The cross section (in blue) after the last bending of a helical twist is rotated by an angle θ about the central axis with respect to the initial cross section (in red). The last two gaskets allow to turn the last cross section (in red) to be a translate of the first one.

In practice, to construct a helical twist of angle $\theta \in (-\pi, \pi]$, we choose an equilateral triangle PQR on the unit sphere, with area θ . Moreover, we fix $P = (1, 0, 0)$, and we take Q in the plane Oxz with positive z coordinate. Then, R is the unique point making PQR equilateral and counterclockwise. Denote by θ_0 the angle between the vectors \vec{OP} and

\overrightarrow{OQ} . By L'Huilier's formula, θ_0 satisfies the equation $4 \arctan \left(\sqrt{\tan(\frac{3}{4}\theta_0) \tan^3(\frac{\theta_0}{4})} \right) = \theta$. Traveling along PQR in trigonometric direction induces a positive rotation angle, while traveling clockwise induces a negative rotation angle. For $|\theta| \leq \pi$, L'Huilier's formula implies $\theta_0 < 1.92$. From (1), we deduce that the corresponding cutting angle satisfies $\lambda < \lambda_0 := 0.89$. Denote by s_0 the initial cross section of the helical twist, by s_1 the cross section at the end of the fourth bend, and by s_2 the initial cross section of the last bend. Refer to Figure 15.

► **Lemma 10.** *Given any twist angle $\theta \in (-\pi, \pi]$ and any $h > 0$, we can construct a helical twist of angle θ so that all its bends have cutting angle λ_0 , and all its gaskets have height h , except the two gaskets between sections s_1 and s_2 , which have height h' imposed by our construction. This helical twist is isometric to a right prism with length*

$$\ell_{twist} = 10a \tan \lambda_0 + 10\bar{h} + 2\bar{h}'$$

and the horizontal distance between the boundaries of the helical twist is bounded by

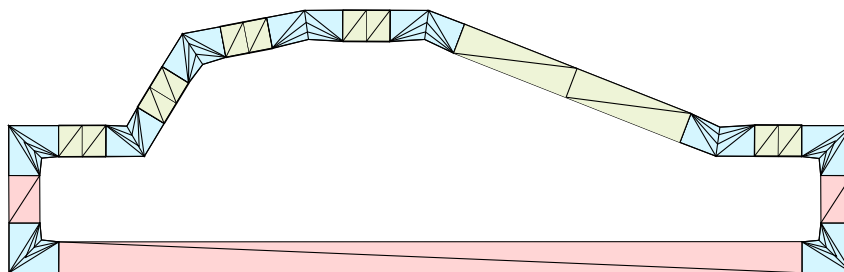
$$d_{twist} = 18(a \tan \lambda_0 + h).$$

Here, \bar{h} and \bar{h}' are given by Equation (2). The height h' is moreover bounded by $2\sqrt{10}(2h + 3a \tan \lambda_0)$.

See the full version for a proof.

Putting the pieces together

Consider a flat torus with modulus $\tau = \tau_1 + i\tau_i$. It can be obtained from a right circular cylinder of height τ_i and boundary length 1, identifying the boundaries after a circular shift at angle $2\pi\tau_1$. Zalgaller constructs his PL isometric embeddings of long tori, for which τ_i is large, as follows. He first replaces the circular cylinder by an isometric equilateral triangular prism that is bent 6 times at angle $\pi/3$ to form a hexagonal tube. If the torus is rectangular, that is if $\tau_1 = 0$, then the identification of the initial and final cross sections provides the desired embedding. Otherwise, he replaces one side of the hexagon by a helical twist of angle $2\pi\tau_1$ in order to glue the boundaries of the prism with the correct angular shift. We use a slightly different construction that allows us to get shorter tori. Starting from a helical twist of angle θ , we add 4 bends at angle $\pi/2$ and 3 portions of right prisms as illustrated on Figure 16 to form a closed torus. In order to avoid intersections between the horizontal prism and the horizontal gaskets of the helical twist we choose the two vertical prisms of



■ **Figure 16** Our construction decomposed into bends (in light blue), gaskets (in light green) and triangular prisms (in pink).

length $\frac{a}{3} > \frac{a}{2\sqrt{3}}$. We also choose the length of the horizontal prism to be equal to the total horizontal extent of the helical twist. We finally take the cutting angle of the 3 bends equals to $\lambda'_0 := \arctan(9/10) > \lambda_0(\pi/2)$. The resulting torus has length

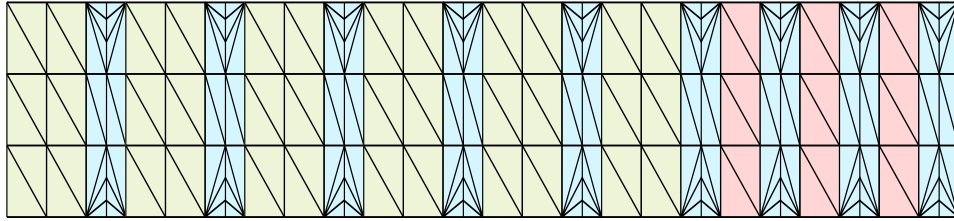
$$L < \ell_{\text{twist}} + 8a \tan \lambda'_0 + 2a/3 + d_{\text{twist}} = 28a \tan \lambda_0 + 8a \tan \lambda'_0 + 2a/3 + 18h + 10\bar{h} + 2\bar{h}',$$

where ℓ_{twist} and d_{twist} are given by Lemma 10. Using the bound for h' in Lemma 10 together with inequality (2), and the fact that $\tan \lambda_0 < 49/40$, and $\tan \lambda'_0 = 9/10$, we get

$$L < \frac{253}{18} + 18h + 10\sqrt{h^2 + \frac{1}{9}} + 2\sqrt{40(2h + \frac{49}{40})^2 + \frac{1}{9}}.$$

By taking $h = 0$ we thus obtain $L < \frac{253}{18} + \frac{10}{3} + 2\sqrt{\frac{49^2}{40} + \frac{1}{9}} < 33$. Note that any longer torus can be obtained by elongating the two vertical prisms. Hence, for h small enough, we can realize any flat torus of length at least 33. In practice, our implementation shows that the same construction allows to embed shorter tori. See Figure 19 in the full version.

We remark that a prism can be triangulated as a gasket with turn 0, the whole construction thus corresponds to the pattern $(g^2b)^5g^2(bg)^3b$ and is composed of $15 \times 6 + 9 \times 20 = 270$ triangles. This ends the proof of Proposition 5. Figure 17 shows the resulting unfolded triangulation after cutting through a cross section and a longitude.



■ Figure 17 A universal triangulation for long tori.

5.2 The flat tori of Tsuboi and Arnoux et al.

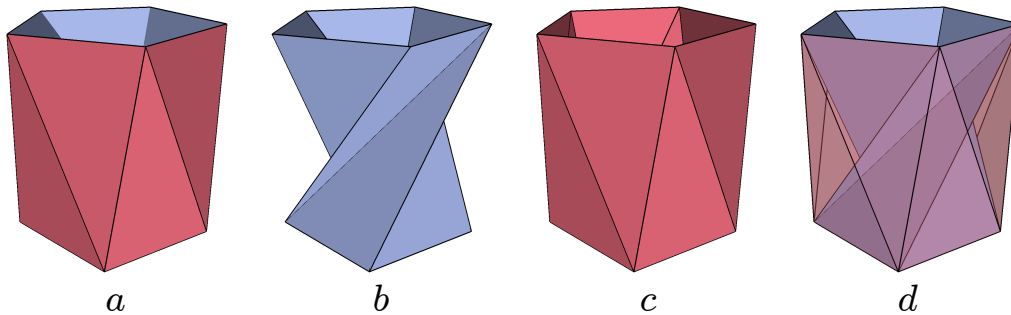
The previous construction provides a universal triangulation for long tori. Referring to Figure 1, this means that the part of the moduli space above the horizontal line $\tau_i = 33$ can be geometrically realized in \mathbb{E}^3 by this unique abstract triangulation. It thus remains to cover the compact subspace of *short tori* below this line. Denote this subspace by $\mathcal{M}_{\text{short}}$. Hence,

$$\mathcal{M}_{\text{short}} = \{\tau \in \mathbb{H}^2 \mid |\tau| \geq 1, |\tau_1| \leq 1/2, |\tau_i| \leq 33\}.$$

As already observed in Section 3, the construction of Burago and Zalgaller allows for some flexibility, implying that around every point in the moduli space there is a neighborhood that can be geometrically realized by the same abstract triangulation. By compactness we can cover $\mathcal{M}_{\text{short}}$ with a finite number of such neighborhoods. We could thus overlay all the corresponding triangulations with the universal triangulation for long tori to obtain a universal triangulation for all tori. This already provides a proof of the existence of such a triangulation. However, estimating the size of the neighborhoods seems impractical and the approach would lead to a gigantic triangulation. Surprisingly, it was only very recently that Tsuboi [9] and Arnoux et al. [1] independently (re)discovered extremely simple geometric

realizations of flat tori. Arnoux et al. are able to prove that their construction, that they call *diploitorus*, allows to realize all tori in the moduli space. For completeness we briefly recall this construction.

The diploitorus $\mathcal{D}_{n,d}^{a,h}$ with parameters n, d, a, h is defined as follows. Let $A_k = (e^{i\frac{2\pi k}{n}}, 0)$ be the vertices of the regular n -gon in the horizontal coordinate plane. Let $B_k = (e^{i\frac{\pi}{n}(a+1+2k)}, h)$ be the vertices of the vertical translate by h of this n -gon, turned by an angle $(a + 1)\frac{\pi}{n}$. Then $\mathcal{D}_{n,d}^{a,h} = \mathcal{P}_{\text{int}} \cup \mathcal{P}_{\text{ext}}$ is the union of two twisted prisms, called *ploids*, where \mathcal{P}_{int} is the union of triangles $\{A_k A_{k+1} B_k\}_{0 \leq k < n}$ and $\{B_k A_{k+1} B_{k+1}\}_{0 \leq k < n}$, and \mathcal{P}_{ext} is the union of triangles $\{A_k A_{k+1} B_{k-d}\}_{0 \leq k < n}$, $\{B_{k-d} A_{k+1} B_{k+1-d}\}_{0 \leq k < n}$. Of course, all the indices should be considered modulo n . Figure 18 shows the diploitorus $\mathcal{D}_{5,2}^{3.5,2}$.



■ **Figure 18** View of the diploitorus $\mathcal{D}_{5,2}^{3.5,2}$ (a) with its internal (b) and external (c) ploids. (d), another view of $\mathcal{D}_{5,2}^{3.5,2}$ with a transparent external ploid.

► **Lemma 11** (Arnoux et al., 2021). For $h, a \in \mathbb{R}$ and $n, d \in \mathbb{Z}$, $\mathcal{D}_{n,d}^{a,h}$ is an embedded flat torus if and only if

$$h > 0, n > 4, 2 \leq |d| < n - 2, d + 1 < a < n - 1 \text{ if } d > 0, \quad \text{and} \quad 1 - n < a < d - 1 \text{ if } d < 0$$

Moreover, the modulus of $\mathcal{D}_{n,d}^{a,h}$ is $\tau(n, d, a, h) = \tau_1(n, d, a) + i\tau_i(n, d, a, h)$ with

$$\tau_1(n, d, a) = d/n - \frac{\cos((a - d)\frac{\pi}{n}) \sin(d\frac{\pi}{n})}{n \sin \frac{\pi}{n}} \quad \text{and}$$

$$\tau_i(n, d, a, h) = \left(\sqrt{h^2 + 4 \sin^2\left(\frac{a+1}{2} \cdot \frac{\pi}{n}\right) \sin^2\left(\frac{a-1}{2} \cdot \frac{\pi}{n}\right)} + \sqrt{h^2 + 4 \sin^2\left(\frac{a-2d+1}{2} \cdot \frac{\pi}{n}\right) \sin^2\left(\frac{a-2d-1}{2} \cdot \frac{\pi}{n}\right)} \right) / (2n \sin(\pi/n))$$

For n, d fixed, we denote by $\mathcal{M}_{n,d}$ the moduli space of the tori $\mathcal{D}_{n,d}^{a,h}$. It lies above the graph of the parametrized curve $a \mapsto (\tau_1(n, d, a), \tau_i(n, d, a, 0))$, where a varies as in Lemma 11.

5.3 Realizing the short tori with three diploitori

The fundamental domain in Figure 1 is symmetric with respect to the imaginary axis. Two symmetric points τ and $-\bar{\tau}$ actually represent isometric tori, but the isometry should reverse the orientation. Hence, if \mathbb{T}_τ has a PL isometric embedding in \mathbb{E}^3 so does $\mathbb{T}_{-\bar{\tau}}$: just take a reflected image of the embedding of \mathbb{T}_τ . It is thus enough to realize the positive part $\mathcal{M}_{\text{short}}^+ := \{\tau \in \mathcal{M}_{\text{short}} \mid \tau_1 \geq 0\}$ of the short tori to ensure that we can realize all of them.

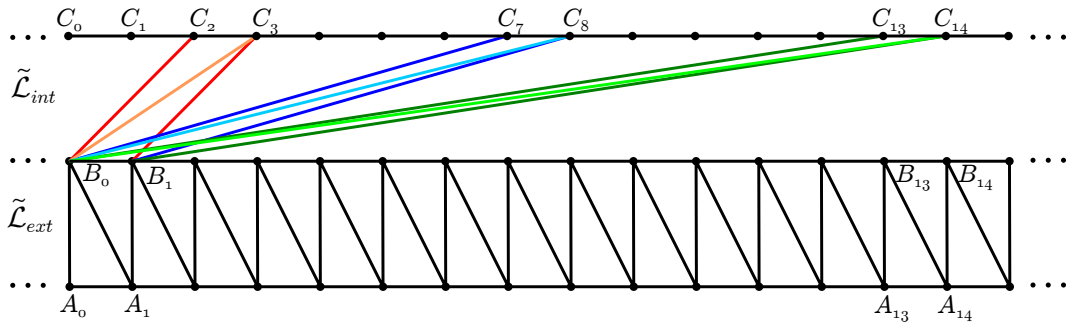
► **Lemma 12.** *Any modulus in \mathcal{M}_{short}^+ can be geometrically realized by a diplotorus with parameters $n = 19$ and $d \in \{2, 7, 13\}$.*

The proof, deferred to the full version, amounts to show that $\mathcal{M}_{19,2} \cup \mathcal{M}_{19,7} \cup \mathcal{M}_{19,13}$ indeed covers all the short tori.

From Lemma 12 we can construct a universal triangulation for short tori. Indeed, all the diplotori with fixed parameters n, d have the same abstract triangulation, that we denote by $\mathcal{T}_{n,d}$. Hence, we just need a common subdivision of $\mathcal{T}_{19,2}$, $\mathcal{T}_{19,7}$ and $\mathcal{T}_{19,13}$ to obtain such a universal triangulation. These triangulations are obtained by identifying the boundaries of a same triangulated cylinder. However, they are not isomorphic, as one needs to apply distinct circular shifts before identification. We can nonetheless send them in a *same* torus as follows. For $k \in \mathbb{Z}$, consider the points

$$A_k = (k, -1), \quad B_k = (k, 0), \quad C_k = (k, 1)$$

in the infinite plane strip $\mathcal{B} := \mathbb{R} \times [-1, 1]$. Then, $\mathcal{T}_{19,d}$ is isomorphic to the triangulation of \mathcal{B} by the triangles $\{A_k A_{k+1} B_k, B_k A_{k+1} B_{k+1}, C_k C_{k+1} B_{k-d}, B_{k-d} C_{k+1} B_{k+1-d}\}_{k \in \mathbb{Z}}$ quotiented by the horizontal translations generated by the vector $(n, 0)$, further identifying the two boundaries according to the vertical translation $(0, 2)$. This quotient and identification being independent of d , the three triangulations for $d = 2, 7, 13$ are indeed embedded in a same torus; see Figure 19.



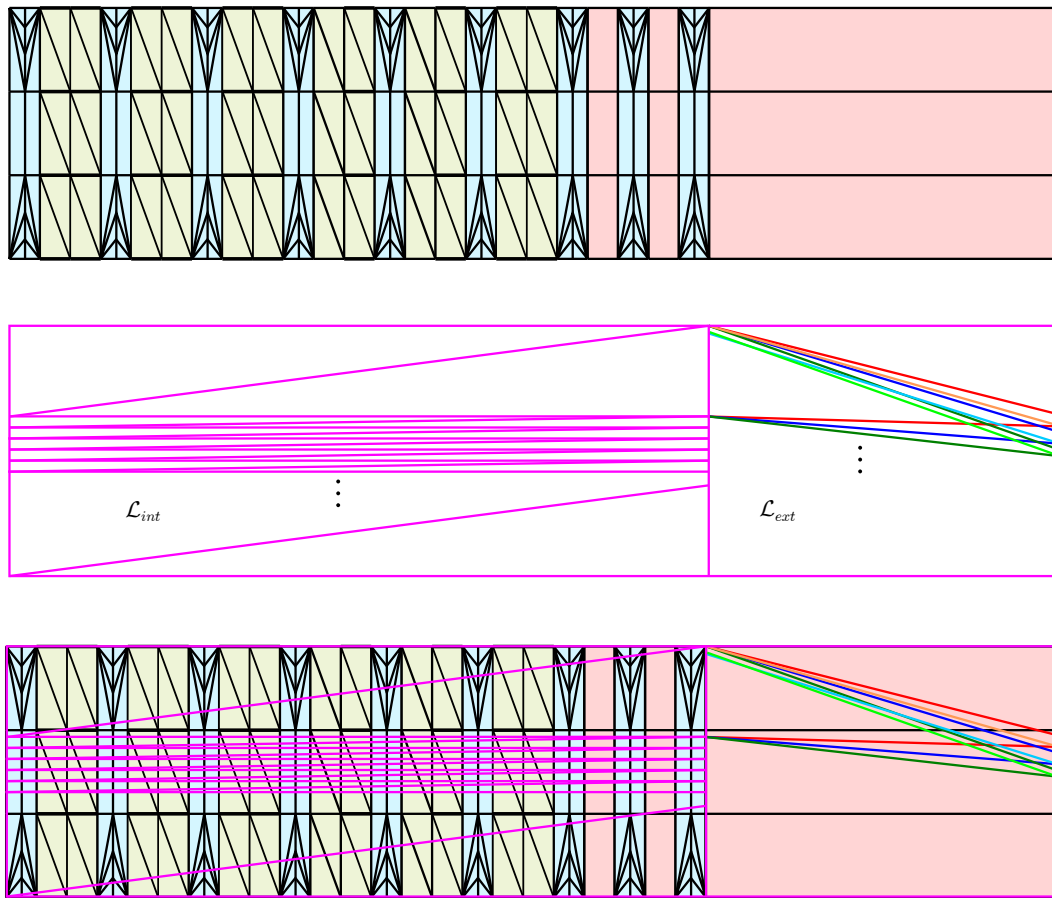
■ **Figure 19** Layout of the triangulations $\mathcal{T}_{19,2}$, $\mathcal{T}_{19,7}$ and $\mathcal{T}_{19,13}$. The two sub-strips $\tilde{\mathcal{L}}_{int}$ and $\tilde{\mathcal{L}}_{ext}$ correspond to the (lift of) overlay of the internal and external ploidis.

We overlay the three triangulated strips obtained for $d = 2, 7, 13$. We can count the number of vertices of the resulting subdivision. We only have to care about the edges $B_k C_L$, the other ones being common to the three triangulations. In the full version we show that these edges intersect in 1064 crossing points. Adding the remaining points A_k, B_k (C_k and A_k should be identified) we find a total of $1064 + 38 = 1102$ vertices. By Euler’s formula on the torus, we conclude that the triangulated overlay has 2204 triangles. We have proved

► **Proposition 13.** *There exists an abstract triangulation with 2204 triangles, which admits linear embeddings isometric to every short torus.*

6 Merging short and long tori

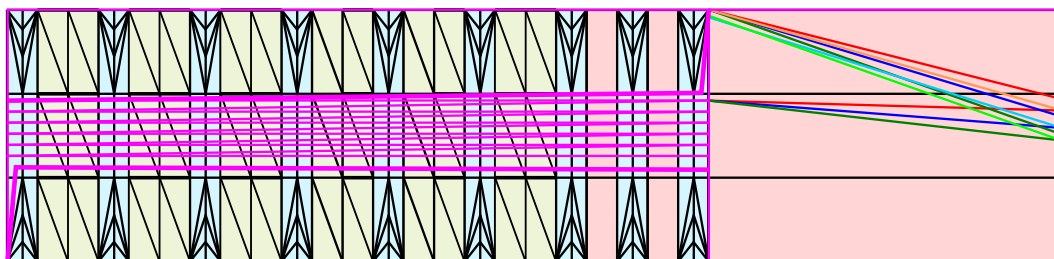
It remains to overlay our universal triangulations for long and short tori to obtain a universal triangulation for all tori. Before overlaying the layouts of Figures 17 and 19, we perform some modifications. We first remove the diagonals introduced to triangulate the rectangular



■ **Figure 20** Modified layout of the universal triangulations for long and short tori, and their overlay.

face of the bends as they are not necessary to define the isometric PL embeddings of long tori. For the same reason, we remove the diagonals used to triangulate the three portions of right prisms; See top Figure 20. Denote by $\mathcal{L}_{\text{long}}$ the resulting layout. We next apply a quarter turn to the layout for short tori, call it $\mathcal{L}_{\text{short}}$. It decomposes into two parts $\mathcal{L}_{\text{int}} \cup \mathcal{L}_{\text{ext}}$ corresponding to the internal and external ploids; See Figure 19. We apply some stretching and compression in order to align \mathcal{L}_{ext} with a portion of right prism in $\mathcal{L}_{\text{long}}$, and to concentrate all the vertices of $\mathcal{L}_{\text{short}}$, except the ones on the horizontal boundaries of the layout, in the central horizontal strip of $\mathcal{L}_{\text{long}}$. We can enumerate the vertices of the overlay as follows. It contains

- $V_{\cap} = 1064$ vertices from the intersecting edges in $\mathcal{L}_{\text{short}}$,
- $3V_{\text{ext}}$ vertices, where V_{ext} is the number of intersections in \mathcal{L}_{ext} of a horizontal edge of $\mathcal{L}_{\text{long}}$ with the edges of $\mathcal{L}_{\text{short}}$,
- $V_{\text{long}} = 270/2 = 135$ vertices from the triangulation of long tori,
- $(2n - 3)V_c = 35V_c$ vertices in the central horizontal strip, where V_c is the number of intersections of an edge of $\mathcal{L}_{\text{short}}$ in this strip with the edges of $\mathcal{L}_{\text{long}}$,
- V_d intersections of the two remaining diagonals of $\mathcal{L}_{\text{short}}$ with the edges of $\mathcal{L}_{\text{long}}$. Rather than considering these two diagonals as line segments, we subdivide each of them by adding a vertex close to their extremities, moving it to the boundary of the central strip; see Figure 21.



■ **Figure 21** The subdivision of the remaining two diagonals of $\mathcal{L}_{\text{long}}$ (thick purple lines).

In the full version of the paper we count $V_{\text{ext}} = 41$, $V_c = 45$, $V_d = 90$, leading to a total of 2987 vertices. By Euler's formula this corresponds, after adding diagonals to triangulate the overlay, to 5974 triangles. This ends the proof of Theorem 1.

Our construction is clearly not optimal. The size of our universal triangulation for long tori can probably be reduced by simplifying the helical twist. The overlay of the triangulation can also be optimized. A challenging question is to find the smallest number of triangles in a universal triangulation for flat tori.

References

- 1 Pierre Arnoux, Samuel Lelièvre, and Alba Málaga. Diplotori: a family of polyhedral flat tori. In preparation, 2021.
- 2 Thomas F. Banchoff. Geometry of the Hopf mapping and Pinkall's tori of given conformal type. In Martin C. Tangora, editor, *Computers in algebra*, volume 111 of *Lecture notes in pure and applied mathematics*, pages 57–62. M. Dekker, 1988.
- 3 Yuriy Dmitrievich Burago and Viktor Abramovich Zalgaller. Isometric piecewise-linear imbeddings of two-dimensional manifolds with a polyhedral metric into \mathbb{R}^3 . *Algebra i analiz*, 7(3):76–95, 1995. Transl. in *St Petersburg Math. J.* (7)3:369–385.
- 4 Jin ichi Itoha and Liping Yuan. Acute triangulations of flat tori. *European journal of combinatorics*, 30:1–4, 2009. doi:10.1016/j.ejc.2008.03.005.
- 5 Nicolaas Kuiper. On C^1 -isometric imbeddings. *Indagationes Mathematicae*, 17:545–555, 1955.
- 6 John F. Nash. C^1 -isometric imbeddings. *Annals of Mathematics*, 60(3):383–396, 1954. doi:10.2307/1969840.
- 7 Ulrich Pinkall. Hopf tori in S^3 . *Inventiones mathematicae*, 81(2):379–386, 1985. doi:10.1007/BF01389060.
- 8 Tanessi Quintanar. An explicit PL-embedding of the square flat torus into \mathbb{E}^3 . *Journal of Computational Geometry*, 11(1):615–628, 2020. doi:10.20382/jocg.v11i1a24.
- 9 Takashi Tsuboi. On origami embeddings of flat tori. *arXiv preprint*, 2020. arXiv:2007.03434.
- 10 V. A. Zalgaller. Some bendings of a long cylinder. *Journal of Mathematical Sciences*, 100(3):2228–2238, 2000. doi:10.1007/s10958-000-0007-3.

Sparse Euclidean Spanners with Tiny Diameter: A Tight Lower Bound

Hung Le ✉

University of Massachusetts, Amherst, MA, USA

Lazar Milenković ✉

Tel Aviv University, Israel

Shay Solomon ✉

Tel Aviv University, Israel

Abstract

In STOC'95 [ADMSS95] Arya et al. showed that any set of n points in \mathbb{R} admits a $(1 + \epsilon)$ -spanner with hop-diameter at most 2 (respectively, 3) and $O(n \log n)$ edges (resp., $O(n \log \log n)$ edges). They also gave a general upper bound tradeoff of hop-diameter at most k and $O(n\alpha_k(n))$ edges, for any $k \geq 2$. The function α_k is the inverse of a certain Ackermann-style function at the $\lfloor k/2 \rfloor$ th level of the primitive recursive hierarchy, where $\alpha_0(n) = \lceil n/2 \rceil$, $\alpha_1(n) = \lceil \sqrt{n} \rceil$, $\alpha_2(n) = \lceil \log n \rceil$, $\alpha_3(n) = \lceil \log \log n \rceil$, $\alpha_4(n) = \log^* n$, $\alpha_5(n) = \lfloor \frac{1}{2} \log^* n \rfloor$, \dots . Roughly speaking, for $k \geq 2$ the function α_k is close to $\lfloor \frac{k-2}{2} \rfloor$ -iterated log-star function, i.e., log with $\lfloor \frac{k-2}{2} \rfloor$ stars. Also, $\alpha_{2\alpha(n)+4}(n) \leq 4$, where $\alpha(n)$ is the one-parameter inverse Ackermann function, which is an extremely slowly growing function.

Whether or not this tradeoff is tight has remained open, even for the cases $k = 2$ and $k = 3$. Two lower bounds are known: The first applies only to spanners with stretch 1 and the second is sub-optimal and applies only to sufficiently large (constant) values of k . In this paper we prove a tight lower bound for any constant k : For any fixed $\epsilon > 0$, any $(1 + \epsilon)$ -spanner for the uniform line metric with hop-diameter at most k must have at least $\Omega(n\alpha_k(n))$ edges.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Euclidean spanners, hop-diameter, inverse-Ackermann, lower bounds, sparse spanners

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.54

Related Version *Full Version*: <https://arxiv.org/abs/2112.09124>

Funding *Hung Le*: Supported by National Science Foundation under Grant No. CCF-2121952.

Lazar Milenković: Partially supported by the Israel Science Foundation (ISF) grant No.1991/1, and by a grant from the United States-Israel Binational Science Foundation (BSF), Israel, and the United States National Science Foundation (NSF).

Shay Solomon: Supported by the Israel Science Foundation (ISF) grant No.1991/1, and by a grant from the United States-Israel Binational Science Foundation (BSF), Israel, and the United States National Science Foundation (NSF).

1 Introduction

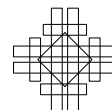
Consider a set S of n points in \mathbb{R}^d and a real number $t \geq 1$. A weighted graph $G = (S, E, w)$ in which the weight function is given by the Euclidean distance, i.e., $w(x, y) = \|x - y\|$ for each $e = (x, y) \in E$, is called a *geometric graph*. We say that a geometric graph G is a t -spanner for S if for every pair $p, q \in S$ of distinct points, there is a path in G between p and q whose *weight* (i.e., the sum of all edge weights in it) is at most t times the Euclidean distance $\|p - q\|$ between p and q . Such a path is called a t -spanner path. The problem of constructing Euclidean spanners has been studied intensively over the years



© Hung Le, Lazar Milenković, and Shay Solomon;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 54; pp. 54:1–54:15
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



[15, 25, 4, 10, 16, 5, 17, 32, 2, 11, 18, 35, 34, 19, 27]. Euclidean spanners are of importance both in theory and in practice, as they enable approximation of the complete Euclidean graph in a more succinct form; in particular, they find a plethora of applications, e.g., in geometric approximation algorithms, network topology design, geometric distance oracles, distributed systems, design of parallel machines, and other areas [16, 28, 32, 20, 22, 21, 23, 29]. We refer the reader to the book by Narasimhan and Smid [30], which provides a thorough account on Euclidean spanners and their applications.

In terms of applications, the most basic requirement from a spanner (besides achieving a small stretch) is to be *sparse*, i.e., to have only a small number of edges. However, for many applications, the spanner is required to preserve some additional properties of the underlying complete graph. One such property, which plays a key role in various applications (such as to routing protocols) [6, 1, 2, 11, 18, 24], is the *hop-diameter*: a t -spanner for S is said to have an hop-diameter of k if, for any $p, q \in S$, there is a t -spanner path between p and q with at most k edges (or hops).

1.1 Known upper bounds

1-spanners for tree metrics. We denote the tree metric induced by an n -vertex (possibly weighted) rooted tree (T, rt) by M_T . A spanning subgraph G of M_T is said to be a *1-spanner* for T , if for every pair of vertices, their distance in G is equal to their distance in T . The problem of constructing 1-spanners for tree metrics is a fundamental one, and has been studied quite extensively over the years, also in more general settings, such as planar metrics [38], general metrics [37] and general graphs [8]. This problem is also intimately related to the extremely well-studied problems of computing partial-sums and online product queries in semigroup and their variants (see [36, 39, 3, 13, 31, 2], and the references therein).

Alon and Schieber [3] and Bodlaender et al. [9] showed that for any n -point tree metric, a 1-spanner with diameter 2 (respectively, 3) and $O(n \log n)$ edges (resp., $O(n \log \log n)$ edges) can be built within time linear in its size. For $k \geq 4$, Alon and Schieber [3] showed that 1-spanners with diameter at most $2k$ and $O(n\alpha_k(n))$ edges can be built in $O(n\alpha_k(n))$ time. The function α_k is the inverse of a certain Ackermann-style function at the $\lfloor k/2 \rfloor$ th level of the primitive recursive hierarchy, where $\alpha_0(n) = \lceil n/2 \rceil$, $\alpha_1(n) = \lceil \sqrt{n} \rceil$, $\alpha_2(n) = \lceil \log n \rceil$, $\alpha_3(n) = \lceil \log \log n \rceil$, $\alpha_4(n) = \log^* n$, $\alpha_5(n) = \lfloor \frac{1}{2} \log^* n \rfloor$, etc. Roughly speaking, for $k \geq 2$ the function α_k is close to $\lfloor \frac{k-2}{2} \rfloor$ -iterated log-star function, i.e., log with $\lfloor \frac{k-2}{2} \rfloor$ stars. Also, $\alpha_{2\alpha(n)+2}(n) \leq 4$, where $\alpha(n)$ is the one-parameter inverse Ackermann function, which is an extremely slowly growing function. (The functions $\alpha_k(n)$ and $\alpha(n)$ are formally defined in [3, 33]; see also Section 2 of the full version [26].) Bodlaender et al. [9] constructed 1-spanners with diameter at most k and $O(n\alpha_k(n))$ edges, with a high running time. Solomon [33] gave a construction that achieved the best of both worlds: a tradeoff of k versus $O(n\alpha_k(n))$ between the hop-diameter and the number of edges in linear time of $O(n\alpha_k(n))$.

Alternative constructions, given by Yao [39] for line metrics and later extended by Chazelle [12] to general tree metrics, achieve a tradeoff of m edges versus $\Theta(\alpha(m, n))$ hop-diameter, where $\alpha(m, n)$ is the standard two-parameter inverse Ackermann function [36]; see also Section 2 of the full version [26]. However, these constructions provide 1-spanners with diameter $\Gamma' \cdot k$ rather than $2k$ or k , for some constant $\Gamma' > 30$.

$(1 + \epsilon)$ -spanners. In STOC'95 Arya et al. [5] proved the so-called ‘‘Dumbbell Theorem’’, which states that, for any d -dimensional Euclidean space, a $(1 + \epsilon, O(\frac{\log(1/\epsilon)}{\epsilon^d}))$ -tree cover can be constructed in $O(\frac{\log(1/\epsilon)}{\epsilon^d} \cdot n \log n + \frac{1}{\epsilon^{2d}} \cdot n)$ time; see Section 2 for the definition of tree cover. The Dumbbell Theorem implies that any construction of 1-spanners for tree metrics can be

translated into a construction of Euclidean $(1 + \epsilon)$ -spanners. Applying the construction of 1-spanners for tree metrics from [33], this gives rise to an optimal $O(n \log n)$ -time construction (in the algebraic computation tree (ACT) model¹) of Euclidean $(1 + \epsilon)$ -spanners. This result can be generalized (albeit not in the ACT model) for the wider family of *doubling metrics*, by using the tree cover theorem of Bartal et al. [7], which generalizes the Dumbbell Theorem of [5] for arbitrary doubling metrics.

1.2 Known lower bounds

The first lower bound on 1-spanners for tree metrics was given by Yao [39] and it establishes a tradeoff of m edges versus hop-diameter of $\Omega(\alpha(m, n))$ for the uniform line metric. Alon and Schieber [3] gave a stronger lower bound on 1-spanners for the uniform line metric: hop-diameter k versus $\Omega(n\alpha_k(n))$ edges, for any k ; it is easily shown that this lower bound implies that of [39] (see Appendix A of the full version [26]), but the converse is not true.

The above lower bounds apply to 1-spanners. There is also a lower bound on $(1 + \epsilon)$ -spanners that applies to line metrics, by Chan and Gupta [11], which extends that of [39]: m edges versus hop-diameter of $\Omega(\alpha(m, n))$. As mentioned already concerning this tradeoff, it only provides a meaningful bound for sufficiently large values of hop-diameter (above say 30), and it does not apply to hop-diameter values that approach 1, which is the focus of this work. More specifically, it can be used to show that any $(1 + \epsilon)$ -spanner for a certain line metric with hop-diameter at most k must have $\Omega(n\alpha_{2k+6}(n))$ edges. When $k = 2$ (resp. $k = 3$), this gives $\Omega(n \log^{****} n)$ (resp. $\Omega(n \log^{*****} n)$) edges, which is far from the upper bound of $O(n \log n)$ (resp., $O(n \log \log n)$). Furthermore, the line metric used in the proof of [11] is not as basic as the uniform line metric – it is derived from hierarchically well-separated trees (HSTs), and to achieve the result for line metrics, an embedding from HSTs to the line with an appropriate separation parameter is employed. The resulting line metric is very far from a uniform one and its aspect ratio² depends on the stretch – it will be super-polynomial whenever ϵ is sufficiently small or sufficiently large; of course, the aspect ratio of the uniform line metric (which is the metric used by [39, 3]) is linear in n . As point sets arising in real-life applications (e.g., for various random distributions) have polynomially bounded aspect ratio, it is natural to ask whether one can achieve a lower bound for a point set of polynomial aspect ratio.

1.3 Our contribution

We prove that any $(1 + \epsilon)$ -spanner for the uniform line metric with hop-diameter k must have at least $\Omega(n\alpha_k(n))$ edges, for any constant $k \geq 2$.

► **Theorem 1.** *For any positive integer n , any integer $k \geq 2$ and any $\epsilon \in [0, 1/2]$, any $(1 + \epsilon)$ -spanner with hop-diameter k for the uniform line metric with n points must contain at least $\Omega(\frac{n}{2^{\lceil k/2 \rceil}} \alpha_k(n))$ edges.*

Interestingly, our lower bound applies also to any $\epsilon > 1/2$, where the bound on the number of edges reduces linearly with ϵ , i.e., it becomes $\Omega(n\alpha_k(n)/\epsilon)$. We stress that our lower bound instance, namely the uniform line metric, does not depend on ϵ , and the lower bound that it provides holds *simultaneously for all values of ϵ* .

¹ Refer to Chapter 3 in [30] for the definition of the ACT model. A matching lower bound of $\Omega(n \log n)$ on the time needed to construct Euclidean spanners is given in [14].

² The *aspect ratio* of a metric is the ratio of the maximum pairwise distance to the minimum one.

Although our lower bound on the number of edges coincides with $\Omega(n\alpha_k(n))$ only for constant k , we note that the values of k of interest range between 1 and $O(\alpha(n))$, where $\alpha(\cdot)$ is a very slowly growing function, e.g., $\alpha(n)$ is asymptotically much smaller than $\log^* n$. Indeed, as mentioned, for $k = 2\alpha(n) + 4$, we have $\alpha_{2\alpha(n)+4}(n) \leq 4$, and clearly any spanner must have $\Omega(n)$ edges. Thus the gap between our lower bound on the number of edges and $\Omega(n\alpha_k(n))$, namely, a multiplicative factor of $2^{6\lfloor k/2 \rfloor}$, which in particular is no greater than $2^{O(\alpha(n))}$, is very small.

For technical reasons we prove a more general lower bound, stated in Theorem 17. In particular, we need to consider a more general notion of Steiner spanners³, and to prove the lower bound for a certain family of line metrics to which the uniform line metric belongs; Theorem 1 follows directly from Theorem 17. See Section 2 for the definitions.

For constant values of k , Theorem 1 strengthens the lower bound shown by [3], which applies only to stretch 1, whereas our tradeoff holds for arbitrary stretch. Whether or not the term $\frac{1}{2^{6\lfloor k/2 \rfloor}}$ in the bound on the number of edges in Theorem 1 can be removed is left open by our work. As mentioned before, we show in Appendix A of the full version [26] that this tradeoff implies the tradeoff by [39] (for stretch 1) and [11] (for larger stretch).

The proof overview appears in the full version [26].

2 Preliminaries

► **Definition 2** (Tree covers). *Let $M_X = (X, \delta_X)$ be an arbitrary metric space. We say that a weighted tree T is a dominating tree for M_X if $X \subseteq V(T)$ and it holds that $\delta_T(x, y) \geq \delta_X(x, y)$, for every $x, y \in X$. For $\gamma \geq 1$ and an integer $\zeta \geq 1$, a (γ, ζ) -tree cover of $M_X = (X, \delta_X)$ is a collection of ζ dominating trees for M_X , such that for every $x, y \in X$, there exists a tree T with $d_T(u, v) \leq \gamma \cdot \delta_X(u, v)$; we say that the stretch between x and y in T is at most γ , and the parameter γ is referred to as the stretch of the tree cover.*

► **Definition 3** (Uniform line metric). *A uniform line metric $U = (\mathbb{Z}, d)$ is a metric on a set of integer points such that the distance between two points $a, b \in \mathbb{Z}$, denoted by $d(a, b)$ is their Euclidean distance, which is $|a - b|$. For two integers $l, r \in \mathbb{Z}$, such that $l \leq r$, we define a uniform line metric on an interval $[l, r]$, denoted by $U(l, r)$, as a subspace of U consisting of all the integer points k , such that $l \leq k \leq r$. We use $U(n)$ to denote a uniform line metric on the interval $[1, n]$.*

Although we aim to prove the lower bound for uniform line metric, the inductive nature of our argument requires several generalizations of the considered metric space and spanner.

► **Definition 4** (t -sparse line metric). *Let l and r be two integers such that $l < r$. We call metric space $U((l, r), t)$ t -sparse if:*

- *It is a subspace of $U(l, r)$.*
- *Each of the consecutive intervals of $[l, r]$ of size t ($[l, l + t - 1], [l + t, l + 2t - 1], \dots$) contains exactly one point. These intervals are called $((l, r), t)$ -intervals and the point inside each such interval is called representative of the interval.*

► **Remark 5.** Throughout the paper, we will always consider Steiner spanners that can contain arbitrary points from the uniform line metric.

³ A Steiner spanner for a point set S is a spanner that may contain additional Steiner points (which do not belong to S). Clearly, a lower bound for Steiner spanners also applies to ordinary spanners.

► **Definition 6** (Global hop-diameter). For any two integers l, r such that $r = l + nt - 1$, let $U((l, r), t)$ be a t -sparse line metric with n points and let X be a subspace of $U((l, r), t)$. An edge that connects two points is $((l, r), t)$ -global if it has endpoints in two different $((l, r), t)$ -intervals of $U((l, r), t)$. A spanner on X with stretch $(1 + \epsilon)$ has its $((l, r), t)$ -global hop-diameter bounded by k if every pair of points in X has a path of stretch at most $(1 + \epsilon)$ consisting of at most k $((l, r), t)$ -global edges.

For ease of presentation, we focus on $\epsilon \in [0, 1/2]$, as this is the basic regime. Our argument naturally extends to any $\epsilon > 1/2$, with the lower bound degrading by a factor of $1/\epsilon$.

► **Lemma 7** (Separation property). Let $l, r, t \in \mathbb{N}$, $l \leq r$, $t \geq 1$ and let $i := \lceil \frac{1+\epsilon/2}{1+\epsilon}l + \frac{\epsilon/2}{1+\epsilon}r \rceil$, and $j := \lfloor \frac{\epsilon/2}{1+\epsilon}l + \frac{1+\epsilon/2}{1+\epsilon}r \rfloor$. Let a, b be two points in $U((l, r), t)$ such that $i \leq a < b \leq j$. Then, any $(1 + \epsilon)$ -spanner path between a and b contains points strictly inside $[l, r]$.

► **Corollary 8**. For every integer $N \geq 34$ and any t -sparse line metric $U((1, N), t)$, any spanner path with stretch at most $3/2$ between metric points a and b such that $\lfloor N/4 \rfloor \leq a \leq b \leq \lceil 3N/4 \rceil$ contains points strictly inside $[1, N]$.

3 Warm-up: lower bounds for hop-diameters 2 and 3

In this section, we prove the lower bound for cases $k = 2$ (Lemma 10 in Section 3.1) and $k = 3$ (Lemma 13 in Section 3.2). In fact, we prove more general statements (Theorems 9 and 12), which apply not only to uniform line metric, but to subspaces of t -sparse line metrics, where a constant fraction of the points is missing. We use these general statements in Section 4, to prove the result for general k (cf. Theorem 17).

3.1 Hop diameter 2

► **Theorem 9**. For any two positive integers $n \geq 1000$ and t , and any two integers l, r such that $r = l + nt - 1$, let $U((l, r), t)$ be a t -sparse line metric with n points and let X be a subspace of $U((l, r), t)$ which contains at least $\frac{31}{32}n$ points. Then, for any choice of $\epsilon \in [0, 1/2]$, any spanner on X with $((l, r), t)$ -global hop-diameter 2 and stretch $1 + \epsilon$ contains at least $T'_2(n) \geq \frac{n}{256} \cdot \alpha_2(n)$ $((l, r), t)$ -global edges which have both endpoints inside $[l, r]$.

The theorem is proved in three steps. First, we prove Lemma 10, which concerns uniform line metrics. Then, we prove Lemma 11 for a subspace that contains at least $31/32$ fraction of the points of the original metric. In the third step, we observe that the same argument applies for t -sparse line metrics.

► **Lemma 10**. For any positive integer n , and any two integers l, r such that $r = l + n - 1$, let $U(l, r)$ be a uniform line metric with n points. Then, for any choice of $\epsilon \in [0, 1/2]$, any spanner on $U(l, r)$ with hop-diameter 2 and stretch $1 + \epsilon$ contains at least $T_2(n) \geq \frac{1}{16} \cdot n \log n$ edges which have both endpoints inside $[l, r]$.

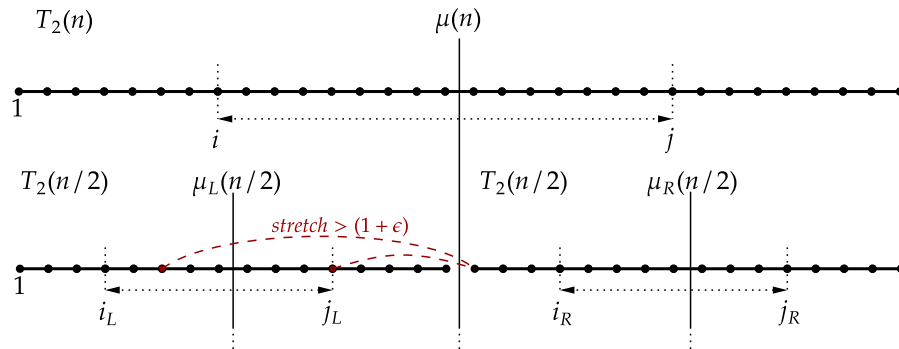
Proof. Suppose without loss of generality that we are working on the uniform line metric $U(1, n)$. Let H be an arbitrary $(1 + \epsilon)$ -spanner for $U(1, n)$ with hop-diameter 2.

For the base case, we take $64 \leq n \leq 127$. In that case our lower bound is $\frac{n}{16} \cdot \log n < n - 1$, which is a trivial lower bound for the number of edges in H , since every two consecutive points must be connected via a direct edge.

For the proof of the inductive step, we can assume that $n \geq 128$. We would like to prove that the number of spanner edges in H is lower bounded by $T_2(n)$, which satisfies recurrence $T_2(n) = 2T_2(\lfloor n/2 \rfloor) + 11n/64$ with the base case $T_2(n) = (n/16) \log n$ when $n \leq 128$. Split

the interval into two disjoint parts: the left part $[1, \lfloor n/2 \rfloor]$ and the right part $[\lfloor n/2 \rfloor + 1, n]$. From the induction hypothesis on the uniform line metric $U(1, \lfloor n/2 \rfloor)$ we know that any spanner with hop-diameter 2 and stretch $1 + \epsilon$ contains at least $T_2(\lfloor n/2 \rfloor)$ edges that have both endpoints inside $[1, \lfloor n/2 \rfloor]$. Similarly, any spanner for $U(\lfloor n/2 \rfloor + 1, n)$ contains at least $T_2(\lfloor n/2 \rfloor)$ edges that have both endpoints inside $[\lfloor n/2 \rfloor + 1, n]$. This means that the sets of edges considered on the left side and the right side are disjoint. We will show below that there are $\Omega(n)$ edges that have one point on the left and the other on the right.

Consider the set L , consisting of the points inside $[n/4, \lfloor n/2 \rfloor]$ and the set R , consisting of the points in $[\lfloor n/2 \rfloor + 1, 3n/4]$. From Corollary 8, since n is sufficiently large, we know that any $(1 + \epsilon)$ -spanner path connecting point $a \in L$ and $b \in R$ has to have all its points inside $[1, n]$. We use term *cross edge* to denote any edge that has one endpoint in the left part and the other endpoint in the right part. We claim that any spanner with hop-diameter at most 2 and stretch $1 + \epsilon$ has to contain at least $\min(|L|, |R|)$ cross edges. For this particular choice of $|L|$ and $|R|$, we have that $\min(|L|, |R|) = |R|$. Suppose for contradiction that the spanner contains less than $|R|$ cross edges. This means that at least one point in $x \in R$ is not connected via a direct edge to any point on the left. Observe that, for every point $l \in L$, the 2-hop spanner path between x and l must be of the form (x, r_l, l) for some point r_l in the right set. It follows that every $l \in L$ induces a different cross edge (r_l, l) . Thus, the number of cross edges, denoted by $|E_C|$, is $|R| \geq |L|$, which is a contradiction. From the definition of L and R , we know that $\min(|L|, |R|) \geq n/4 - 2$, implying that the number of cross edges is at least $n/4 - 2 \geq 11n/64$, for all $n \geq 26$. (See also Figure 1 for an illustration.) Thus, we have: $T_2(n) = 2T_2(\lfloor n/2 \rfloor) + \frac{11n}{64} \geq 2 \cdot \frac{\lfloor n/2 \rfloor}{16} \log \lfloor n/2 \rfloor + \frac{11n}{64} \geq \frac{n}{16} \cdot \log n$ as claimed. ◀



■ **Figure 1** An illustration of the first two levels of the recurrence for the lower bound for $k = 2$ and $\epsilon = 1/2$. We split the interval $U(1, n)$ into two disjoint parts. In Lemma 10, we show that there will be at least $\Omega(n)$ cross edges, which are the spanner edges having endpoints in both parts. The values i_L and j_L are set according to Corollary 8 so that the spanner edges crossing $\mu(n)$ cannot be used for the left set; otherwise the resulting stretch will be bigger than $1 + \epsilon$.

► **Lemma 11** (Proof omitted; see the full version [26]). *For any positive integer n , and any two integers l, r such that $r = l + n - 1$, let $U(l, r)$ be a uniform line metric with n points and let X be a subspace of $U(l, r)$ which contains at least $\frac{31}{32}n$ points. Then, for any choice of $\epsilon \in [0, 1/2]$, any spanner on X with hop-diameter 2 and stretch $1 + \epsilon$ contains at least $T'_2(n) \geq 0.48 \cdot \frac{n}{16} \log n$ edges which have both endpoints inside $[l, r]$.*

Completing the proof of Theorem 9. Note that $\alpha_2(n) = \lceil \log n \rceil$ and hence, we will show that $T'_2(n) \geq \frac{n}{256} \lceil \log n \rceil$. Suppose without loss of generality that we are working on any t -sparse line metric with n points, $U((1, N), t)$, where $N = nt$. Let H be an arbitrary $(1 + \epsilon)$ -spanner for $U((1, N), t)$ with $((1, N), t)$ -global hop-diameter 2. We would like to lower bound the number of $((l, r), t)$ -global edges required for H .

Since $\epsilon \in [0, 1/2]$, every two consecutive points in $U((1, N), t)$, except for the leftmost and the rightmost two, have to be connected by a spanner path which has all its endpoints inside the interval $[1, N]$. This implies that the number of spanner edges is at least $n - 3$, which is in turn greater than $(n/16) \log n$, for any $64 \leq n \leq 127$.

Let $M = \lfloor n/2 \rfloor t$ and let L be the set of $((l, r), t)$ -intervals that are fully inside $[N/4, M]$ and R be the set of $((l, r), t)$ -intervals that are fully inside $[M, 3N/4]$. In that case, the number of $((l, r), t)$ -intervals inside L can be lower bounded by $|L| \geq \lfloor (M - N/4 + 1)/t \rfloor \geq n/4 - 2$, which is the bound that we used for L . Similarly, we obtain that $|R| \geq n/4 - 1$. The cross edges will be those edges that contain one endpoint in $[1, M]$ and the other endpoint in $[M + 1, N]$. It follows that the cross edges are also $((l, r), t)$ -global edges. The same argument can be applied to lower bound the number of cross edges, implying the lower bound on the number of $((l, r), t)$ -global edges. The same proof as in Lemma 11 gives $T'_2(n) \geq 0.48 \cdot \frac{n}{16} \log n \geq \frac{n}{256} \lceil \log n \rceil$, when $n \geq 1000$, as desired. ◀

3.2 Hop diameter 3

► **Theorem 12.** *For any two positive integers $n \geq 1000$ and t , and any two integers l, r such that $r = l + nt - 1$, let $U((l, r), t)$ be a t -sparse line metric with n points and let X be a subspace of $U((l, r), t)$ which contains at least $\frac{127}{128}n$ points. Then, for any choice of $\epsilon \in [0, 1/2]$, any spanner on X with $((l, r), t)$ -global hop-diameter 3 and stretch $1 + \epsilon$ contains at least $T'_3(n) \geq \frac{n}{1024} \cdot \alpha_3(n)$ $((l, r), t)$ -global edges which have both endpoints inside $[l, r]$.*

The theorem is proved in three steps. First, we prove Lemma 13, which concerns uniform line metrics. Then, we prove Lemma 16 for a subspace that contains at least $31/32$ fraction of the points of the original metric. In the third step, we observe that the same argument applies for t -sparse line metrics.

► **Lemma 13.** *For any positive integer n , and any two integers l, r such that $r = l + n - 1$, let $U(l, r)$ be a uniform line metric with n points. Then, for any choice of $\epsilon \in [0, 1/2]$, any spanner on $U(l, r)$ with hop-diameter 3 and stretch $1 + \epsilon$ contains at least $T_3(n) \geq \frac{n}{40} \log \log n$ edges which have both endpoints inside $[l, r]$.*

Proof. Suppose without loss of generality that we are working on the uniform line metric $U(1, n)$. Let H be an arbitrary $(1 + \epsilon)$ -spanner for $U(1, n)$ with hop-diameter 3.

For the base case, we assume that $11 \leq n \leq 127$. We have that $\frac{n}{40} \log \log n < n - 1$, which is a trivial lower bound on the number of edges of H , since every two consecutive points have to be connected via a direct edge.

We now assume that $n \geq 128$. Divide the the interval $[1, n]$ into consecutive subintervals containing $b := \lfloor \sqrt{n} \rfloor$ points: $[1, b], [b + 1, 2b]$, etc. Our goal is to show that the number of spanner edges is lower bounded by $T_3(n)$, which satisfies recurrence $T_3(n) = \left\lfloor \frac{n}{\lfloor \sqrt{n} \rfloor} \right\rfloor \cdot T_3(\lfloor \sqrt{n} \rfloor) + n/18$, with the base case $T_3(n) = (n/40) \log \log n$ when $n < 128$.

For any j such that $1 \leq j \leq \lfloor n/b \rfloor$, the interval spanned by the j th subinterval is $[(j - 1)b + 1, jb]$. Using the induction hypothesis, any spanner on $U((j - 1)b + 1, jb)$ contains at least $T_3(b)$ edges that are inside $[(j - 1)b + 1, jb]$. This means that all the subintervals

will contribute at least $\lfloor n/b \rfloor \cdot T_3(b)$ spanner edges that are mutually disjoint and in addition do not go outside of $[1, n]$. We will show that there are $\Omega(n)$ edges that have endpoints in two different subintervals, called *cross edges*. By definition, the set of cross edges is disjoint from the set of spanner edges considered in the term $\lfloor n/b \rfloor \cdot T_3(b)$.

Consider the points that are within interval $[n/4, 3n/4]$. From Corollary 8, since n is sufficiently large, we know that any $(1 + \epsilon)$ -spanner path connecting two points in $[n/4, 3n/4]$ has to have all its points inside $[1, n]$.

We call a point *global* if it is adjacent to at least one cross edge. Otherwise, the point is *non-global*. The following two claims bound the number of cross edges induced by global and non-global points, respectively.

▷ **Claim 14.** Suppose that among points inside interval $[n/4, 3n/4]$, m of them are global. Then, they induce at least $m/2$ spanner edges.

The claim is true since each global point contributes at least one cross edge and each edge is counted at most twice.

▷ **Claim 15.** Suppose that among points inside interval $[n/4, 3n/4]$, m of them are non-global. Then, they induce at least $\binom{m/\sqrt{n}}{2}$ cross edges.

Proof. Consider two sets A and B such that A contains a non-global point $a \in [n/4, 3n/4]$ and B contains a non-global point $b \in [n/4, 3n/4]$. Since a is non-global, it can be connected via an edge either to a point inside of A or to a point outside of $[1, n]$. Similarly, b can be connected to either a point inside of B or to a point outside of $[1, n]$. From Corollary 8, and since $a, b \in [n/4, 3n/4]$, we know that every spanner path with stretch $(1 + \epsilon)$ connecting a and b has to use points inside $[1, n]$. This means that the spanner path with stretch $(1 + \epsilon)$ has to have a form (a, a', b', b) , where $a' \in A$ and $b' \in B$. In other words, we have to connect points a' and b' using a cross edge; furthermore every pair of intervals containing at least one non-global point induce one such edge and for every pair this edge is different.

Each interval contains at most $b = \lfloor \sqrt{n} \rfloor$ non-global points, so the number of sets containing at least one non-global point is at least m/b . Interconnecting all the sets requires $\binom{m/b}{2} \geq \binom{m/\sqrt{n}}{2}$ edges. ◁

The number of points inside $[n/4, 3n/4]$ is at least $n/2 + 1$, but we shall use a slightly weaker lower bound of $15n/32$. We consider two complementary cases. In the first case, at least $1/4$ of $15n/32$ points are global. Claim 14 implies that the number of the cross edges induced by these points is at least $15n/256$. The other case is that at least $3/4$ fraction of $15n/32$ points are non-global. Claim 15 implies that for a sufficiently large n , the number of cross edges induced by these points can be lower bounded by $15n/256$ as well. In other words, we have shown that in both cases, the number of cross edges is at least $\frac{15}{256}n > \frac{n}{18}$. Thus, we have: $T_3(n) \geq \left\lfloor \frac{n}{\lfloor \sqrt{n} \rfloor} \right\rfloor \cdot T_3(\lfloor \sqrt{n} \rfloor) + \frac{n}{18} \geq \lfloor \sqrt{n} \rfloor \cdot \frac{\lfloor \sqrt{n} \rfloor}{40} (\log \log \lfloor \sqrt{n} \rfloor) + \frac{n}{18}$, which is at most $\frac{n}{40} \log \log n$, as claimed. ◀

► **Lemma 16** (Proof omitted; see the full version [26]). *For any positive integer n , and any two integers l, r such that $r = l + n - 1$, let $U(l, r)$ be a uniform line metric with n points and let X be a subspace of $U(l, r)$ which contains at least $\frac{127}{128}n$ points. Then, for any choice of $\epsilon \in [0, 1/2]$, any spanner on X with hop-diameter 3 and stretch $1 + \epsilon$ contains at least $T'_3(n) \geq 0.18 \cdot \frac{n}{40} \log \log n$ edges which have both endpoints inside $[l, r]$.*

Completing the proof of Theorem 12. Note that $\alpha_3(n) = \lceil \log \log n \rceil$ and hence, we will show that $T'_3(n) \geq \frac{n}{1024} \cdot \lceil \log \log n \rceil$. Suppose without loss of generality that we are working on any t -sparse line metric with n points, $U((1, N), t)$, where $N = nt$. Let H be an arbitrary $(1 + \epsilon)$ -spanner for $U((1, N), t)$ with $((1, N), t)$ -global hop-diameter 3. We would like to lower bound the number of $((l, r), t)$ -global edges required for H .

Since $\epsilon \in [0, 1/2]$, every two consecutive points in $U((1, N), t)$, except for the leftmost and the rightmost two, have to be connected by a spanner path which has all its endpoints inside the interval $[1, N]$. This implies that the number of spanner edges is at least $n - 3$, which is in turn greater than $(n/40) \log \log n$, for any $11 \leq n \leq 127$.

Let consider the set of $((l, r), t)$ -intervals that are fully inside $[N/4, 3N/4]$. The number of such intervals can be lower bounded by $((3N/4 - N/4)/t - 2 \geq n/2 - 2$, which is larger than the bound of $15n/32$, which we used. The cross edges will become $((1, N), t)$ -global edges and the same argument can be applied to lower bound their number. The same proof in Lemma 16 gives:

$$T'_3(n) \geq 0.18 \cdot \frac{n}{40} \log \log n \geq \frac{n}{1024} \cdot \lceil \log \log n \rceil$$

when $n \geq 1000$, as desired. ◀

4 Lower bound for constant hop-diameter

We proceed to prove our main result, which is a generalization of Theorem 1. In particular, invoking Theorem 17 stated below where X is the uniform line metric $U(1, n)$ gives Theorem 1.

► **Theorem 17.** *For any two positive integers $n \geq 1000$ and t , and any two integers l, r such that $r = l + nt - 1$, let $U((l, r), t)$ be a t -sparse line metric with n points and let X be a subspace of $U((l, r), t)$ which contains at least $n(1 - \frac{1}{2^{k+4}})$ points. Then, for any choice of $\epsilon \in [0, 1/2]$ and any integer $k \geq 2$, any spanner on X with $((l, r), t)$ -global hop-diameter k and stretch $1 + \epsilon$ contains at least $T'_k(n) \geq \frac{n}{2^{6\lceil k/2 \rceil + 4}} \cdot \alpha_k(n)$ $((l, r), t)$ -global edges which have both endpoints inside $[l, r]$.*

Proof. We will prove the theorem by double induction on $k \geq 2$ and n . The base case for $k = 2$ and $k = 3$ and every n is proved in Theorems 9 and 12, respectively.

For every $k \geq 4$, we shall prove the following two assertions.

1. For any two positive integers n and t , and any two integers l, r such that $r = l + nt - 1$, let $U((l, r), t)$ be a t -sparse line metric with n points. Then, for any choice of $\epsilon \in [0, 1/2]$, any spanner on $U((l, r), t)$ with $((l, r), t)$ -global hop-diameter k and stretch $1 + \epsilon$ contains at least $T_k(n) \geq \frac{n}{2^{6\lceil k/2 \rceil + 2}} \alpha_k(n)$ $((l, r), t)$ -global edges which have both endpoints inside $[l, r]$.
2. For any two positive integers n and t , and any two integers l, r such that $r = l + nt - 1$, let $U((l, r), t)$ be a t -sparse line metric with n points and let X be a subspace of $U((l, r), t)$ which contains at least $n(1 - \frac{1}{2^{k+4}})$ points. Then, for any choice of $\epsilon \in [0, 1/2]$, any spanner on X with $((l, r), t)$ -global hop-diameter k and stretch $1 + \epsilon$ contains at least $T'_k(n) \geq \frac{n}{2^{6\lceil k/2 \rceil + 4}} \cdot \alpha_k(n)$ $((l, r), t)$ -global edges which have both endpoints inside $[l, r]$.

For every $k \geq 4$, we first prove the first assertion, which relies on the second assertion for $k - 2$. Then, we prove the second assertion which relies on the first assertion for k . We proceed to prove assertion 1.

Proof of assertion 1. Suppose without loss of generality that we are working on any t -sparse line metric $U((1, N), t)$. Let H be an arbitrary $(1 + \epsilon)$ -spanner for $U((1, N), t)$ with $((1, N), t)$ -global hop-diameter k .

Let M be $A((k-2)/2, 4)$ if k is even and $B(\lfloor (k-2)/2 \rfloor, 4)$ if k is odd. For the base case take $4 \leq n < \max(M, 10000)$. We consider $n-2$ points in $U((1, N), t)$: all the points from the metric, excluding the leftmost and the rightmost one. Since $\epsilon \in [0, 1/2]$, every two consecutive points among the considered $n-2$ points have to be connected by a spanner path which has all its endpoints inside the interval $[1, N]$. This implies that the number of spanner edges is at least $n-3$. Then $\frac{n}{2^{6\lfloor k/2 \rfloor + 2}} \alpha_k(n)$, which is at most $\frac{n}{2^{6\lfloor k/2 \rfloor + 2}} \log^*(n) \leq n-3$.

Next, we prove the induction step. We shall assume the correctness of the two statements: (i) for k and all smaller values of n , and (ii) for $k' < k$ and all values of n . Let $N := nt$ and let $b := \alpha_{k-2}(n)$. Divide the interval $[1, N]$ into consecutive $((1, N), bt)$ -intervals containing b points: $[1, bt], [bt+1, 2bt]$, etc. We would like to prove that the number of spanner edges is lower bounded by recurrence

$$T_k(n) = \left\lfloor \frac{n}{\alpha_{k-2}(n)} \right\rfloor \cdot T_k(\alpha_{k-2}(n)) + \frac{n}{2^{6\lfloor k/2 \rfloor + 1}},$$

with the base case $T_k(n) = \frac{n}{2^{6\lfloor k/2 \rfloor + 2}} \alpha_k(n)$ for $n \leq 10000$.

There are $\lfloor n/b \rfloor$ $((1, N), bt)$ -intervals containing exactly b points. For any j such that $1 \leq j \leq \lfloor n/b \rfloor$, the j th $((1, N), bt)$ -interval is $[(j-1)bt+1, jbt]$. Using inductively the assertion 1 for k and a value $b < n$, any spanner on $U((j-1)bt+1, jbt)$ contains at least $T_k(b)$ edges that are inside $[(j-1)bt+1, jbt]$. This means that all the $((1, N), bt)$ -intervals will contribute at least $\lfloor n/b \rfloor \cdot T_k(b)$ spanner edges that are mutually disjoint and in addition do not go outside of $[1, N]$.

We will show that there are $\Omega(n/2^{3k})$ edges that have endpoints in two different $((1, N), bt)$ -intervals, i.e. edges that are $((1, N), bt)$ -global. Since these edges are $((1, N), bt)$ -global, they are disjoint from the spanner edges considered in the term $\lfloor n/b \rfloor \cdot T_k(b)$. We shall focus on points that are inside $((1, N), bt)$ -intervals fully inside $[N/4, 3N/4]$; denote the number of such points by p . We have $p \geq n/2 - 2\alpha_{k-2}(n)$, but we will use a weaker bound:

$$p \geq n/4. \tag{1}$$

► **Definition 18.** A point that is incident on at least one $((1, N), bt)$ -global edge is called a $((1, N), bt)$ -global point.

Among the p points inside $[N/4, 3N/4]$, denote by p' the number of $((1, N), bt)$ -global points. Let $p'' = p - p'$, and m be the number of $((1, N), bt)$ -global edges incident on the p points. Since each $((1, N), bt)$ -global point contributes at least one $((1, N), bt)$ -global edge and each such edge is counted at most twice, we have

$$m \geq p'/2. \tag{2}$$

Next, we prove that

$$m \geq \frac{n}{2^{6\lfloor k/2 \rfloor + 1}}, \quad \text{if } \left\lceil \frac{p''}{b} \right\rceil \geq \left(1 - \frac{1}{2^{k+2}}\right) \cdot \left\lceil \frac{p}{b} \right\rceil \tag{3}$$

Recall that we have divided $[1, N]$ into consecutive $((1, N), bt)$ -intervals containing $b := \alpha_{k-2}(n)$ points. Consider now all the $((1, N), bt)$ -intervals that are fully inside $[N/4, 3N/4]$, and denote this collection of $((1, N), bt)$ -intervals by \mathcal{C} . Let l' (resp. r') be the leftmost (resp. rightmost) point of the leftmost (resp. rightmost) interval in \mathcal{C} ; note that l' and r' may not coincide with points of the input metric, they are simply the leftmost and rightmost boundaries of the intervals in \mathcal{C} .

Constructing a new line metric. For each $((1, N), bt)$ -interval I in \mathcal{C} , if I contains a point that is not $((1, N), bt)$ -global, assign an arbitrary such point in I as its representative; otherwise, assign an arbitrary point as its representative. The collection \mathcal{C} of $((1, N), bt)$ -intervals, together with the set of representatives uniquely defines (bt) -sparse line metric, $U((l', r'), bt)$. This metric has $\lceil p/b \rceil$ $((1, N), bt)$ -intervals, since there are $\lceil p/b \rceil$ intervals covering p points in the input t -sparse metric $U((1, N), t)$ inside the interval $[N/4, 3N/4]$. Recall from Definition 4 that a bt -sparse metric is uniquely defined given its $((1, N), bt)$ -intervals and representatives. Let X be the subspace of $U((l', r'), bt)$ induced by the representatives of all intervals in \mathcal{C} that contain points that are not $((1, N), bt)$ -global and using Equation (3), we have

$$|X| \geq \left\lceil \frac{p''}{b} \right\rceil \geq \left(1 - \frac{1}{2^{k+2}}\right) \cdot \left\lceil \frac{p}{b} \right\rceil \tag{4}$$

Recall that H is an arbitrary $(1 + \epsilon)$ -spanner for $U((1, N), t)$ with $((1, N), t)$ -global hop-diameter k . Let a and b be two arbitrary points in X , and denote their corresponding $((1, N), bt)$ -intervals by A and B , respectively. Since a (reps., b) is not $((1, N), bt)$ -global, it can be adjacent either to points outside of $[1, N]$ or to points inside A (resp., B). By Corollary 8 and since $a, b \in [N/4, 3N/4]$, any spanner path with stretch $(1 + \epsilon)$ connecting a and b must remain inside $[1, N]$. Hence, any $(1 + \epsilon)$ -spanner path in H between a and b is of the form (a, a', \dots, b', b) , where $a' \in A$ (resp. $b' \in B$). Consider now the same path in the metric X . It has at most k hops, where the first and the last edges are not $((1, N), bt)$ -global. Thus, although this path contains at most k $((1, N), t)$ -global edges in $U((1, N), t)$, it has at most $k - 2$ $((1, N), bt)$ -global edges in X . It follows that H is a (Steiner) $(1 + \epsilon)$ -spanner with $((1, N), bt)$ -global hop-diameter $k - 2$ for X . See Figure 2 for an illustration.

Denote by $n' := \lceil p/b \rceil$ the number of points in $U((l', r'), bt)$. Since $p \geq n/4$, it follows that $n' \geq \lceil n/(4b) \rceil$. By (4), X is a subspace of $U((l', r'), bt)$, and its size is at least a $(1 - 1/2^{k+2})$ -fraction (i.e., a $(1 - 1/2^{(k-2)+4})$ -fraction) of that of $U((l', r'), bt)$. Hence, by the induction hypothesis of assertion 2 for $k - 2$, we know that any spanner on X with $((l', r'), bt)$ -global hop-diameter $k - 2$ and stretch $1 + \epsilon$ contains at least $T'_{k-2}(n') \geq \frac{n'}{2^{6\lfloor (k-2)/2 \rfloor + 4}} \cdot \alpha_{k-2}(n')$ $((l', r'), bt)$ -global edges which have both endpoints inside $[l', r']$. Since every $((l', r'), bt)$ -global edge is also a $((1, N), bt)$ -global edge, we conclude with the following lower bound on the number of $((1, N), bt)$ -global edges required by H :

$$\begin{aligned} T'_{k-2}(n') &\geq \frac{n'}{2^{6\lfloor (k-2)/2 \rfloor + 4}} \cdot \alpha_{k-2}(n') \\ &\geq \frac{n}{4 \cdot 2^{6\lfloor (k-2)/2 \rfloor + 4} \cdot \alpha_{k-2}(n)} \cdot \alpha_{k-2}\left(\left\lceil \frac{n}{4\alpha_{k-2}(n)} \right\rceil\right) \\ &\geq \frac{n}{8 \cdot 2^{6\lfloor (k-2)/2 \rfloor + 4}} \\ &= \frac{n}{2^{6\lfloor k/2 \rfloor + 1}} \end{aligned}$$

The last inequality follows since, when $k \geq 4$, the ratio between $\alpha_{k-2}(\lceil n/4\alpha_{k-2}(n) \rceil)$ and $\alpha_{k-2}(n)$ can be bounded by $1/2$ for sufficiently large n (i.e. larger than the value considered in the base case). In other word, we have shown that whenever $\lceil p''/b \rceil \geq (1 - 1/2^{k+2}) \cdot \lceil p/b \rceil$, the number of the $((1, N), bt)$ -global edges incident on the p points inside $[N/4, 3N/4]$ is lower bounded by $n/2^{6\lfloor k/2 \rfloor + 1}$; we have thus proved (3).

Recall (see (1)) that we lower bounded the number p of points inside $[N/4, 3N/4]$ as $p \geq n/4$. We consider two complementary cases: either $\lceil p''/b \rceil \geq (1 - 1/2^{k+2}) \cdot \lceil p/b \rceil$, or $\lceil p''/b \rceil < (1 - 1/2^{k+2}) \cdot \lceil p/b \rceil$, where p'' is the number of points in $[N/4, 3N/4]$ that are not $((1, N), bt)$ -global. In the former case (i.e. when $\lceil p''/b \rceil \geq (1 - 1/2^{k+2})$), by (3), we have

the number of $((1, N), bt)$ -global edges is lower bounded by $n/2^{6\lfloor k/2 \rfloor + 1}$. In the latter case, we have $\frac{p-p'}{b} - 1 < \lfloor \frac{p-p'}{b} \rfloor = \frac{p''}{b} < (1 - \frac{1}{2^{k+2}}) \cdot \lceil \frac{p}{b} \rceil < (1 - \frac{1}{2^{k+2}}) \cdot \frac{p}{b} + 1$. In other words, we can lower bound p' by $p/2^{k+2} - 2b$. From (2) and using that $p \geq n/4$, the number of $((1, N), bt)$ -global edges is lower bounded by $n/2^{k+5} - \alpha_{k-2}(n)$. Since the former bound is always smaller for n sufficiently large (i.e. larger than the value considered in the base case), we shall use it as a lower bound on the number of $((1, N), bt)$ -global edges required by H . We note that every $((1, N), bt)$ -global edge is also $((1, N), t)$ -global, as required by assertion 1. It follows that

$$\begin{aligned} T_k(n) &\geq \left\lfloor \frac{n}{\alpha_{k-2}(n)} \right\rfloor \cdot \frac{\alpha_{k-2}(n)}{2^{6\lfloor k/2 \rfloor + 2}} \cdot \alpha_k(\alpha_{k-2}(n)) + \frac{n}{2^{6\lfloor k/2 \rfloor + 1}} \\ &\geq \left(\frac{n}{\alpha_{k-2}(n)} - 1 \right) \cdot \frac{\alpha_{k-2}(n)}{2^{6\lfloor k/2 \rfloor + 2}} \cdot (\alpha_k(n) - 1) + \frac{n}{2^{6\lfloor k/2 \rfloor + 1}} \\ &\geq \frac{n}{2^{6\lfloor k/2 \rfloor + 2}} \alpha_k(n) \end{aligned}$$

For the second inequality we have used that $\alpha_k(n) = 1 + \alpha_k(\alpha_{k-2}(n))$, and for the third, the fact that $\alpha_{k-2}(n) \cdot (\alpha_k(n) - 1) \leq n$ for sufficiently large n (i.e. larger than the value considered in the base case). This concludes the proof of assertion 1.

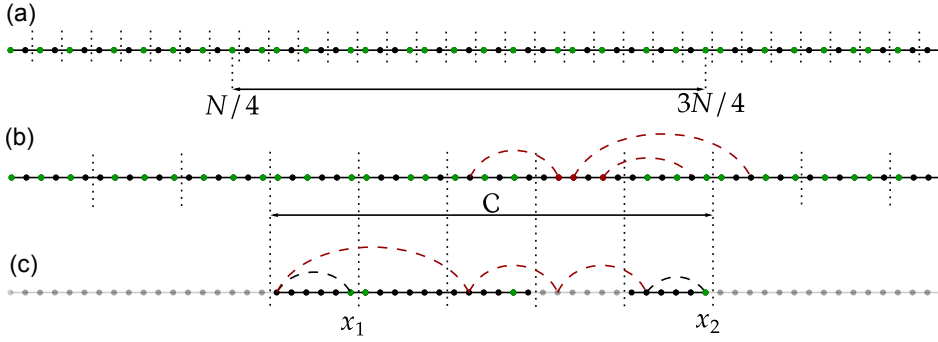


Figure 2 Constructing a new line metric and invoking the induction hypothesis. **(a)** We have $n = 32$, $k = 5$, and a 2-sparse line metric $U((1, 64), 2)$ with representatives of each $((1, 64), 2)$ -interval highlighted in green. **(b)** Since $b = \alpha_{k-2}(n) = 3$, we consider a collection of $((1, 64), 6)$ -global intervals inside $[N/4, 3N/4]$, denoted by \mathcal{C} . The seventh block contains only $((1, 64), 6)$ -global points (highlighted in red) as each of them is incident on a $((1, 64), 6)$ -global edge. **(c)** The new line metric is 6-sparse line metric $U((19, 48), 6)$ consisting of 4 green points. Finally, we use the induction hypothesis of assertion 2 for $k = 3$ to lower bound the number of $((1, N), 6)$ -global edges. A spanner path between x_1 and x_2 consisting of 5 edges, 3 of which are $((1, N), 6)$ global is depicted.

Proof of assertion 2. Suppose without loss of generality that we are working on any t -sparse line metric $U((1, N), t)$. Let H be an arbitrary $(1 + \epsilon)$ -spanner for $U(1, N)$ with $((1, N), t)$ -global hop-diameter k . We shall inductively assume the correctness of assertion 1 and assertion 2: (i) for k and all smaller values of n , and (ii) for $k' < k$ and all values of n .

Recall the recurrence we used in the proof of assertion 1, $T_k(n) = \lfloor n/\alpha_{k-2}(n) \rfloor \cdot T_k(\alpha_{k-2}(n)) + \frac{n}{2^{6\lfloor k/2 \rfloor + 1}}$, which provides a lower bound on the number of $((l, r), t)$ -global edges of H . The base case for this recurrence is whenever $n < 10000$. Consider the recursion tree of $T_k(n)$ and denote its depth by ℓ and the number of nodes at depth i by c_i . In addition, denote by $n_{i,j}$ the number of points in the j th interval of the i th level and by $e_{i,j}$ the number of $((1, N), t)$ -global edges contributed by this interval. We have that the contribution of an interval is $n_{i,j}/2^{6\lfloor k/2 \rfloor + 1}$. By definition, we have $T_k(n) = \sum_{i=1}^{\ell} \sum_{j=1}^{c_i} e_{i,j} \geq \frac{n}{2^{6\lfloor k/2 \rfloor + 2}} \alpha_k(n)$.

Let H' be any $(1 + \epsilon)$ spanner on X with $((1, N), t)$ -global hop-diameter k . To lower bound the number of spanner edges in H' , we now consider the same recursion tree, but take into consideration the fact that we are working on metric X , which is a subspace of $U((1, N), t)$. This means that at each level of recursion, instead of n points, there is at least $n(1 - 1/2^{k+4})$ points in X . The contribution of the j th interval in the i th level is denoted by $e'_{i,j}$. We call the j th interval in the i th level *good* if it contains at least $n_{i,j}(1 - 1/2^{k+3})$ points from X . (Recall that we have used $n_{i,j}$ to denote the number of points from $U(l, r)$ in the j th interval of the i th level.) From the definition of good interval and the fact that each level of recurrence contains at least $n(1 - 1/2^{k+4})$ points, it follows that there are at least $n/2$ points contained in the good intervals at the i th level. Denote the collection of all the good intervals at the i th level by Γ_i .

Recall that we are working with recurrence $T_k(n) = \lfloor n/\alpha_{k-2}(n) \rfloor \cdot T_k(\alpha_{k-2}(n)) + \frac{n}{2^{6\lfloor k/2 \rfloor + 1}}$. In particular, in the first level of recurrence, we consider the contribution of n points, whereas in the second level, we consider the contribution of $\lfloor n/\alpha_{k-2}(n) \rfloor \cdot \alpha_{k-2}(n)$ points. Denote by n_i the number of points whose contribution we consider in the i th level of recurrence. Then, we have $n_1 = n$, $n_2 = \lfloor n/\alpha_{k-2}(n) \rfloor \cdot \alpha_{k-2}(n) \geq n - \alpha_{k-2}(n)$. Denote by $\alpha_{k-2}^{(j)}(n)$ value of $\alpha_{k-2}(\cdot)$ iterated on n , i.e. $\alpha_{k-2}^{(0)}(n) = n$, $\alpha_{k-2}^{(1)}(n) = \alpha_{k-2}(n)$, $\alpha_{k-2}^{(2)}(n) = \alpha_{k-2}(\alpha_{k-2}(n))$, etc. In general, for $i \geq 2$, we have $n_i \geq n - \sum_{j=2}^i \frac{n\alpha_{k-2}^{(j-1)}(n)}{\alpha_{k-2}^{(j-2)}(n)} \geq n - n \cdot \sum_{j=2}^i \frac{\lceil \log^{(j-1)}(n) \rceil}{\lceil \log^{(j-2)}(n) \rceil}$. We observe that there is an exponential decay between the numerator and denominator of terms in each summand and that terms grow with j . Since we do not consider intervals in the base case, we also know that $\lceil \log^{(i-1)}(n) \rceil \geq 10000$, meaning that the largest term in the sum is $10000/2^{9999}$. By observing that every two consecutive terms increase by a factor larger than 2, we conclude that $n_i \geq 0.99n$. Since at each level there are at least $n/2$ points inside of good intervals, this means that there are at least $0.49n$ points inside of good intervals which were not ignored. Denote by Γ_i the set of good intervals in the i th level whose contribution is not ignored. Then we have $T'_k(n) = \sum_{i=1}^{\ell} \sum_{j=1}^{c'_i} e'_{i,j} \geq \sum_{i=1}^{\ell} \sum_{j \in \Gamma_i} e_{i,j} \geq 0.49 \cdot T_k(n) \geq \frac{n}{2^{6\lfloor k/2 \rfloor + 4}} \alpha_k(n)$. This concludes the proof of assertion 2. We have thus completed the inductive step for k . ◀

References

- 1 Ittai Abraham and Dahlia Malkhi. Compact routing on euclidian metrics. In *PODC*, pages 141–149. ACM, 2004.
- 2 Pankaj K. Agarwal, Yusu Wang, and Peng Yin. Lower bound for sparse euclidean spanners. In *SODA*, pages 670–671. SIAM, 2005.
- 3 Noga Alon and Baruch Schieber. Optimal preprocessing for answering on-line product queries. Technical report, Tel Aviv University, 1987.
- 4 Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discret. Comput. Geom.*, 9:81–100, 1993.
- 5 Sunil Arya, Gautam Das, David M. Mount, Jeffrey S. Salowe, and Michiel H. M. Smid. Euclidean spanners: short, thin, and lanky. In *STOC*, pages 489–498. ACM, 1995.
- 6 Sunil Arya, David M. Mount, and Michiel H. M. Smid. Randomized and deterministic algorithms for geometric spanners of small diameter. In *FOCS*, pages 703–712. IEEE Computer Society, 1994.
- 7 Yair Bartal, Nova Fandina, and Ofer Neiman. Covering metric spaces by few trees. In *ICALP*, volume 132 of *LIPICs*, pages 20:1–20:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- 8 Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners. In *SODA*, pages 932–941. SIAM, 2009.

- 9 Hans L. Bodlaender, Gerard Tel, and Nicola Santoro. Trade-offs in non-reversing diameter. *Nord. J. Comput.*, 1(1):111–134, 1994.
- 10 Paul B. Callahan and S. Rao Kosaraju. Faster algorithms for some geometric graph problems in higher dimensions. In *SODA*, pages 291–300. ACM/SIAM, 1993.
- 11 T.-H. Hubert Chan and Anupam Gupta. Small hop-diameter sparse spanners for doubling metrics. *Discret. Comput. Geom.*, 41(1):28–44, 2009.
- 12 Bernard Chazelle. Computing on a free tree via complexity-preserving mappings. *Algorithmica*, 2:337–361, 1987.
- 13 Bernard Chazelle and Burton Rosenberg. The complexity of computing partial sums off-line. *Int. J. Comput. Geom. Appl.*, 1(1):33–45, 1991.
- 14 Danny Z. Chen, Gautam Das, and Michiel H. M. Smid. Lower bounds for computing geometric spanners and approximate shortest paths. *Discret. Appl. Math.*, 110(2-3):151–167, 2001.
- 15 Paul Chew. There is a planar graph almost as good as the complete graph. In *SCG*, pages 169–177. ACM, 1986.
- 16 Gautam Das and Giri Narasimhan. A fast algorithm for constructing sparse euclidean spanners. *Int. J. Comput. Geom. Appl.*, 7(4):297–315, 1997.
- 17 Gautam Das, Giri Narasimhan, and Jeffrey S. Salowe. A new way to weigh malnourished euclidean graphs. In *SODA*, pages 215–222. ACM/SIAM, 1995.
- 18 Yefim Dinitz, Michael Elkin, and Shay Solomon. Low-light trees, and tight lower bounds for euclidean spanners. *Discret. Comput. Geom.*, 43(4):736–783, 2010.
- 19 Michael Elkin and Shay Solomon. Optimal euclidean spanners: Really short, thin, and lanky. *J. ACM*, 62(5):35:1–35:45, 2015.
- 20 Joachim Gudmundsson, Christos Levcopoulos, Giri Narasimhan, and Michiel H. M. Smid. Approximate distance oracles for geometric graphs. In *SODA*, pages 828–837. ACM/SIAM, 2002.
- 21 Joachim Gudmundsson, Christos Levcopoulos, Giri Narasimhan, and Michiel H. M. Smid. Approximate distance oracles for geometric spanners. *ACM Trans. Algorithms*, 4(1):10:1–10:34, 2008.
- 22 Joachim Gudmundsson, Giri Narasimhan, and Michiel H. M. Smid. Fast pruning of geometric spanners. In *STACS*, volume 3404 of *Lecture Notes in Computer Science*, pages 508–520. Springer, 2005.
- 23 Yehuda Hassin and David Peleg. Sparse communication networks and efficient routing in the plane. *Distributed Comput.*, 14(4):205–215, 2001.
- 24 Omri Kahalon, Hung Le, Lazar Milenkovic, and Shay Solomon. Can’t see the forest for the trees: Navigating metric spaces by bounded hop-diameter spanners. *CoRR*, abs/2107.14221, 2021. [arXiv:2107.14221](https://arxiv.org/abs/2107.14221).
- 25 J. Mark Keil and Carl A. Gutwin. Classes of graphs which approximate the complete euclidean graph. *Discret. Comput. Geom.*, 7:13–28, 1992.
- 26 Hung Le, Lazar Milenkovic, and Shay Solomon. Sparse euclidean spanners with tiny diameter: A tight lower bound. *CoRR*, abs/2112.09124, 2021. [arXiv:2112.09124](https://arxiv.org/abs/2112.09124).
- 27 Hung Le and Shay Solomon. Truly optimal euclidean spanners. In *FOCS*, pages 1078–1100. IEEE Computer Society, 2019.
- 28 Christos Levcopoulos, Giri Narasimhan, and Michiel H. M. Smid. Efficient algorithms for constructing fault-tolerant geometric spanners. In *STOC*, pages 186–195. ACM, 1998.
- 29 Yishay Mansour and David Peleg. An approximation algorithm for minimum-cost network design. In *Robust Communication Networks: Interconnection and Survivability*, volume 53 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 97–106. DIMACS/AMS, 1998.
- 30 Giri Narasimhan and Michiel H. M. Smid. *Geometric spanner networks*. Cambridge University Press, 2007.
- 31 Mihai Patrascu and Erik D. Demaine. Tight bounds for the partial-sums problem. In *SODA*, pages 20–29. SIAM, 2004.

- 32 Satish Rao and Warren D. Smith. Approximating geometrical graphs via “spanners” and “banyans”. In *STOC*, pages 540–550. ACM, 1998.
- 33 Shay Solomon. Sparse euclidean spanners with tiny diameter. *ACM Trans. Algorithms*, 9(3):28:1–28:33, 2013.
- 34 Shay Solomon. From hierarchical partitions to hierarchical covers: optimal fault-tolerant spanners for doubling metrics. In *STOC*, pages 363–372. ACM, 2014.
- 35 Shay Solomon and Michael Elkin. Balancing degree, diameter and weight in euclidean spanners. In *ESA (1)*, volume 6346 of *Lecture Notes in Computer Science*, pages 48–59. Springer, 2010.
- 36 Robert Endre Tarjan. Applications of path compression on balanced trees. *J. ACM*, 26(4):690–715, 1979.
- 37 Mikkel Thorup. On shortcutting digraphs. In *WG*, volume 657 of *Lecture Notes in Computer Science*, pages 205–211. Springer, 1992.
- 38 Mikkel Thorup. Shortcutting planar digraphs. *Comb. Probab. Comput.*, 4:287–315, 1995.
- 39 Andrew Chi-Chih Yao. Space-time tradeoff for answering range queries (extended abstract). In *STOC*, pages 128–136. ACM, 1982.

Minimum Height Drawings of Ordered Trees in Polynomial Time: Homotopy Height of Tree Duals

Tim Ophelders  

Department of Information and Computing Science, Utrecht University, The Netherlands
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Salman Parsa  

Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, USA

Abstract

We consider drawings of graphs in the plane in which vertices are assigned distinct points in the plane and edges are drawn as simple curves connecting the vertices and such that the edges intersect only at their common endpoints. There is an intuitive quality measure for drawings of a graph that measures the height of a drawing $\phi: G \hookrightarrow \mathbb{R}^2$ as follows. For a vertical line ℓ in \mathbb{R}^2 , let the height of ℓ be the cardinality of the set $\ell \cap \phi(G)$. The height of a drawing of G is the maximum height over all vertical lines. In this paper, instead of abstract graphs, we fix a drawing and consider plane graphs. In other words, we are looking for a homeomorphism of the plane that minimizes the height of the resulting drawing. This problem is equivalent to the homotopy height problem in the plane, and the homotopic Fréchet distance problem. These problems were recently shown to lie in NP, but no polynomial-time algorithm or NP-hardness proof has been found since their formulation in 2009. We present the first polynomial-time algorithm for drawing trees with optimal height. This corresponds to a polynomial-time algorithm for the homotopy height where the triangulation has only one vertex (that is, a set of loops incident to a single vertex), so that its dual is a tree.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Graph drawing, homotopy height

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.55

Related Version *Full Version:* <https://arxiv.org/abs/2203.08364>

Funding *Tim Ophelders:* This author was supported by the Dutch Research Council (NWO) under project no. VI.Veni.212.260.

Salman Parsa: This author was funded in part by the SLU Research Institute and by NSF grant CCF-1614562.

1 Introduction

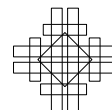
A tree T is called an *ordered tree* if for each vertex, a fixed cyclic ordering of its incident edges is given. Let T be an ordered tree and let $f: |T| \rightarrow \mathbb{R}^2$ be a drawing of the tree, that is, a continuous injection from the underlying topological space of the tree to the plane, in which the clockwise order of edges around each vertex is as prescribed. Any ordered tree can be recovered from any of its drawings up to degree 2 nodes. Any two drawings of the same ordered tree can be obtained from one another using an orientation-preserving homeomorphism of the plane. We are interested in drawings that minimize the height in the following sense. Given a drawing ϕ and a vertical line ℓ , the *height* of the line ℓ is defined as $H(\ell) := |\phi(T) \cap \ell|$. That is, the number of times that the line ℓ intersects the drawing, where vertical segments count as infinitely many intersections. The problem of drawing a tree T with optimal height asks for a drawing $\phi: |T| \rightarrow \mathbb{R}^2$ that minimizes the maximum height over all vertical lines. We call such a drawing an *optimal height drawing*. We emphasize that our drawings are not necessarily straight-line. In fact, there exist instances for which any

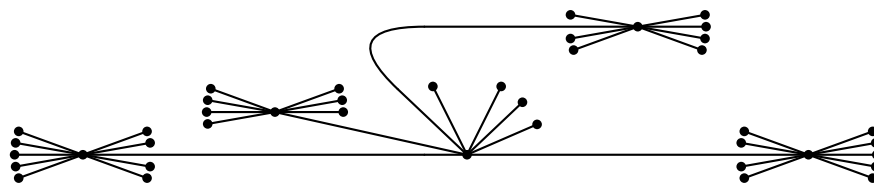


© Tim Ophelders and Salman Parsa;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 55; pp. 55:1–55:16
Leibniz International Proceedings in Informatics

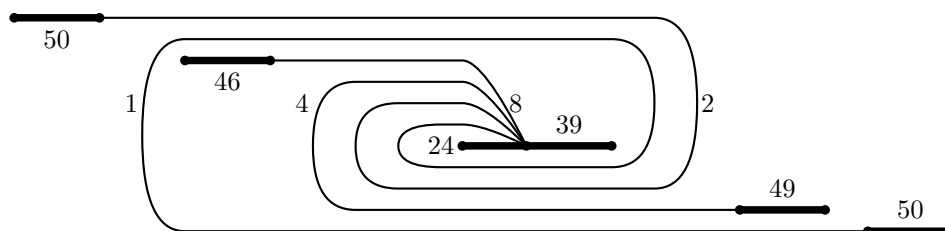


LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** A bend is necessary in any drawing with height 5.

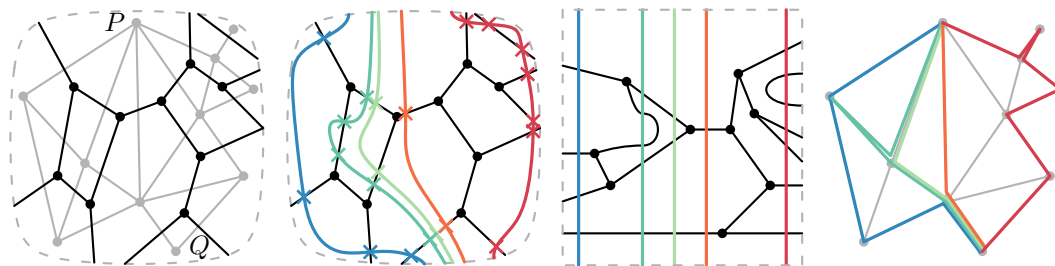


■ **Figure 2** Spirals (e.g. the edge with weight 1) may be necessary to draw weighted trees optimally.

optimal drawing requires a bend in some edge. An example is given in Figure 1. One can check that any optimal drawing of this tree requires a bend in some edge. Although we will consider only unweighted trees, the definition of height naturally extends to edge-weighted graphs. Already in the case of weighted trees with only one vertex of degree at least three, an optimal drawing might even require an edge to form a spiral. Figure 2 depicts an instance whose optimal drawing requires a spiral according to a computer-assisted enumeration of its drawings. We do not know whether unweighted trees also require spiraling edges.

The optimal height drawing of graphs is related to two significant classes of problems in computer science, and in particular, computational geometry and topology. If, instead of ordered trees, we take (unordered) trees and allow edges to cross in the output drawing, we obtain the classical min-cut linear arrangement problem. This problem is well-studied [7, 11, 14] and Yannakakis [15] presented an $O(n \log n)$ time algorithm for drawing trees with optimal height in this sense. Of course, optimal drawings with straight-line edges always exist in this setting. On the other hand, it is known that the graph version as well as the weighted tree version [13] of the same problem is NP-hard. Since the trees corresponding to the reduction can be drawn optimally without self-intersection, it follows that optimal height drawing of unordered weighted trees is also NP-hard. All the mentioned problems lie in NP.

The optimal height graph drawing problem also shows up as a special case of an important open problem in computational geometry and topology called the homotopy height problem [2, 3, 5, 6, 10]. In this context, a homotopy corresponds to a one-parameter family of curves γ_i ($i \in [0, 1]$) that sweeps a surface in a continuous way, where γ_0 and γ_1 are part of the input. Roughly speaking, the homotopy height problem considers a surface homeomorphic to a sphere, disk, or annulus, endowed with a metric, and asks for a homotopy of curves that sweeps the surface in such a way that the longest curve γ_i is as short as possible. For a perfectly round sphere, the homotopy height is the length of its equator. For the purpose of computation, discrete versions of the problem have been considered, where the surface is endowed with a cellular decomposition, and the lengths of curves are measured by the number of intersections with cell boundaries. Each curve in general position with the cellular decomposition can be represented as a walk on the dual graph of the decomposition. The vertices of the dual graph are represented geometrically as representative points of cells, and edges of the dual graph correspond to pieces of cell boundaries shared by two cells. As a curve



■ **Figure 3** Left: a cellular decomposition of a disk (black) and its dual. Middle: some curves of a homotopy whose curves start at P and end at Q , and a homeomorphism of the disk that sends the curves to vertical lines. Right: the corresponding walks in the dual graph.

sweeps over the surface, it can sweep over vertices of the dual graph (resulting in a face flip), or create or remove pairs of intersections with edges of the embedded graph (resulting in a spike or unspike). Figure 3 illustrates a dual graph (of a cellular decomposition) with vertices P and Q , and a homotopy through curves γ_i connecting P to Q . If the cell decomposition contains exactly one vertex, then its dual is a tree and the problem of homotopy height becomes equivalent to drawing trees with optimal height (In this case, the starting and ending curves are nested circles in the unbounded face so that the curves sweep an annulus).

Although homotopy height admits an efficient $O(\log n)$ -approximation algorithm [10], its exact computation appears to be very challenging. In fact, it was only recently shown to lie in the complexity class NP [5] in the setting of edge-weighted graphs. If the curves at the start and end of a homotopy are disjoint and shortest curves, it is known that there exists an optimal homotopy that sweeps the surface in a monotone fashion [4]. Homotopy height is closely related to other important graph parameters [2].

The duality relation between graph drawings and homotopy height is depicted in Figure 3.

In this paper instead of graphs we consider plane trees or ordered trees. We present the first polynomial-time algorithm for the optimal height drawing of unit weight plane trees. Our results give a polynomial algorithm for the homotopy height of unit-weight one-vertex (multi-)graphs. This might point to the possibility that the problem for the general graphs is also polynomial. However, already in our restricted setting, the algorithm is quite involved and does not have a clear extension to general graphs.

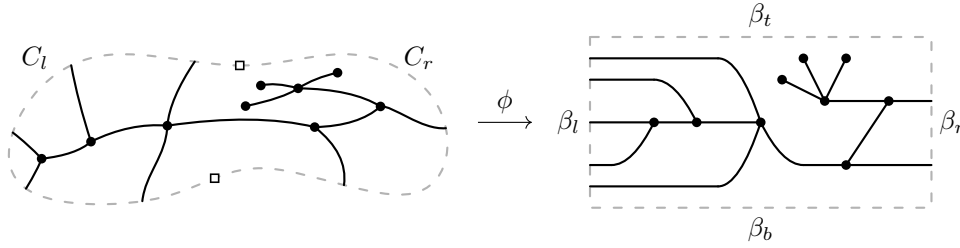
Although our notion of height has frequently been studied in recent years, there exist related parameters of graph drawings that also quantify some notion of height [1, 2, 12].

2 Background and terminology

2.1 Drawings and local disks

Drawings. Formally we work with *plane trees* instead of ordered trees. This is just some reasonable, e.g. piecewise-linear, drawing $g: T \rightarrow \mathbb{R}^2$ of a finite tree T in the Euclidean plane. This plane drawing is fixed once and for all for any ordered tree T and respects the given ordering around each vertex. In order to distinguish the Euclidean plane containing the drawing g we use the symbol Π for this plane, so that $g(T) \subset \Pi$.

Convention. With a slight abuse of notation, we will not distinguish between T and its embedding $g(T) \subset \Pi$ in the plane. We use the words edge and path for edges and simple curves on T exclusively, and reserve the word curve for curves in the drawing plane (the plane to which Π is mapped).



■ **Figure 4** A local disk and a drawing.

A *drawing* ϕ of a tree T , is a continuous injective function mapping Π into \mathbb{R}^2 . We consider only drawings ϕ in which the image of every edge e is piecewise-linear, and such that every vertical line intersects the drawing in a finite number of points. It is not difficult to see that this restriction does not affect the optimal height of the drawing. In our figures, for aesthetic purposes, we often draw edges as smooth curves.

Let $E = E(T)$, $V = V(T)$ denote the set of edges and vertices of T . We always denote the number of vertices by n . By $H(\phi)$ we denote the *height* of the drawing ϕ . That is, the maximum number of points of the drawing on a vertical line.

Local disks. Let $D \subset \Pi(T)$ be a topological disk in the plane in which T is drawn. We denote the boundary of D by ∂D . Let $T_D = T \cap D$ and assume T_D is connected. We say that an edge $e \in E$ is a *boundary edge* of D if $e \cap \partial D \neq \emptyset$. We call an edge *internal* if it lies in the interior of D . We denote by $B(D)$ the set of boundary edges of D . Let $\partial D = C_l \cup C_r$ where $C_l \cap C_r = \{p_N, p_S\}$ is a set of two points, where none is in T . Intuitively, we think of C_l and C_r as the left and right boundary of D . This “partition” of ∂D divides the set of boundary edges $B(D)$ into *left* and *right boundary edges* $B(D) = B_L(D) \sqcup B_R(D)$. We call $(D, B_L(D), B_R(D))$ a *local disk*.

A *drawing of a local disk* (D, B_L, B_R) is a homeomorphism $\phi: D \rightarrow Q$ onto a rectangle Q with edges $(\beta_l, \beta_t, \beta_r, \beta_d)$, such that under ϕ , the boundary edges in B_L intersect β_l , and those in B_R intersect β_r and $\phi(T_D) \cap (\beta_t \cup \beta_r) = \emptyset$. See Figure 4. Note that we can select a local disk whose interior contains the whole tree, such that $T_D = T$ and there are no boundary edges. The height of the left (right) boundary in any drawing is the number of left (right) boundary edges of the local disk, and we call this number as the *left (right) boundary height*. When the two boundary heights are equal we simply say *boundary height*.

The move sequence of a drawing. Consider sweeping a vertical line over a drawing of T (or the interior of a local disk). The sweep line encounters three types of events: left bends (points interior to edges of T whose x -coordinate in the drawing is locally minimal), right bends (symmetric to left bends), and vertices. We will refer to these events as *moves*, and the corresponding point of T as its *location* (i.e. the vertex corresponding to a vertex move, or the point interior to the edge corresponding to the bend move). We assume that all bends and vertex moves occur at distinct x -coordinates, and refer to the left-to-right sequence of moves of a drawing as its *move sequence*.

2.2 Cuts and shortcuts

Let D be a local disk. By a *cut* in the local disk (D, B_L, B_R) we mean the sequence of edges crossed by a curve that connects p_N to p_S , where p_N and p_S are some two points giving rise to the local disk (D, B_L, B_R) (an edge might repeat consecutively in the sequence). Some

times we refer to the curve itself as a cut. Note that the same local disk can be defined with many such pair of points but this choice is not important. The *length* (or *height*) of a cut is the number of edges in it (counted with repetition), or the number of intersections of the curve with the tree T_D . A cut C is a *shortcut* if its length is smallest over all cuts of D . For the proof of the following lemma we refer to [5, Lemma 4.2].

► **Lemma 1** (Pausing at a shortcut). *Let D be a local disk, $\phi: D \rightarrow Q$ a drawing and C a shortcut in D . There is a drawing ϕ' of height less than or equal to the height of ϕ in which there is a vertical line defining the cut C . Moreover, vertical lines of ϕ that are disjoint from C are unaffected and appear in ϕ' .*

We say that the drawing ϕ can *pause* at the shortcut C , resulting in the drawing ϕ' . When a cut C is vertical in an optimal drawing and each sub-disk cut by C contains a connected part of T_D , then C subdivides the problem into two sub-problems whose optimal drawings can easily be merged to form an optimal drawing of the original disk.

3 Overview of the algorithm

Our main result is an algorithm for computing optimal drawings of plane trees. This algorithm is a dynamic program which in a high level works as follows. Each cell of the dynamic programming table represents a local disk and stores the optimal height of that disk (or an optimal drawing, if an optimal drawing is to be computed). The local disks represented by the cells are of two special types: spine disks and skew spine disks (defined in Section 6). These disks essentially are local disks that cannot be cut by shortcuts. Row m of the table stores all spine or skew spine disks with exactly m interior vertices. For $m > 1$, row m of the table is built using the information in lower rows in two phases. The first phase constructs all possible m -vertex spine and skew spine disks. The second phase computes the height of an optimal height drawing for each of the computed disks of row m (or computes a drawing, if the the drawing is needed). The computed optimal height (or optimal drawing) will be stored again in the table. The base of the table consists of spine or skew spine disks with a single interior vertex. The possibilities for the decomposition of a single spine disk or skew spine disk into such disks with fewer vertices are shown in Figures 10 and 11. With this description, a final optimal drawing consists of drawings in Figures 10 and 11 nested inside each other, and where the deepest level is a single vertex disk. A trapezoid (skew spine disk) will fit into a trapezoid and a rectangle into a rectangle (spine disk).

There are two ingredients in the proof of correctness. First, we show in Proposition 6 that any (sufficiently general) drawing can be turned, without increasing the height, into a drawing that has a hierarchical structure. The root of this structure tree is a spine disk containing the whole tree (with zero boundary edges). The nodes of the structure tree are (skew) spine disks. Each node is cut essentially into a collection of sub-disks, using shortcuts that are made vertical via pausing. These sub-disks are (skew) spine disks that form the children of the node. Each node, has one of polynomially many possibilities for the decomposition, depicted in Figures 10 and 11. The spine disks corresponding to leaves of any structure tree are single-vertex local disks and thus trivial to draw optimally. In brief, any drawing can be turned into one which has a tree structure of spine and skew spine disks without increasing the height.

The second ingredient of the proof of correctness is a proof that there exists some optimal drawing such that a super-set of all the (skew) spine disks in its tree structure can be enumerated in polynomial time. For this purpose, we define a quality measure for a drawing. To rule out pathological drawings and simplify our arguments, we need to consider simplified

drawings which are the result of applying simplification moves of Figure 6. We also consider balanced drawings, which are ones where the height of the lines on both sides of (and very close to) any vertex differ by at most one. Among all the optimal drawings, we take the drawing which is simplified, balanced, and maximizes our quality measure. Lemma 7 asserts that such an optimized drawing has itself a tree structure of (skew) spine disks. The tree structure of a drawing which optimizes a slightly stronger measure, namely the secondary quality, is called a fat structure. Proposition 10 characterizes the spine disks that can appear in a fat structure. This description allows us to easily enumerate all possible spine disks that can appear in a fat structure in polynomial time and only store the (skew) spine disks in our table that conform to this characterization. This will result in a polynomial-sized table and hence a polynomial algorithm. Omitted proofs can be found in the full version.

4 Simplifying the drawings

Let ϕ be a drawing of a local disk D and $T = T_D$. We label the left (resp. right) bends of ϕ as either *stuck* or not, depending on whether the bend encloses the next (resp. previous) move. Figure 5 illustrates the two possible reasons for a right bend to be stuck. Two consecutive moves of a drawing may admit a simplification (of the drawing) that replaces the two sequences by a simpler sequence of moves without increasing the height of the drawing. For each of these simplifications, either the first move is a non-stuck left bend, or the second move is a non-stuck right bend. We explain the types of simplifications involving a non-stuck right bend (see Figure 6), the types involving a non-stuck left bend are symmetric. As mentioned, the second move is a non-stuck right bend, so we distinguish cases based on the first move of the pair.

Stuck slide. In this case, the first move is a stuck right bend. The non-stuck right bend does not enclose the stuck right bend (otherwise it would also be stuck). Exchanging the order of the two bends ensures that neither of the resulting bends are stuck.

Bend-bend (resp. vertex-bend) separation. The first move is a left bend (resp. vertex) that is not connected to the right bend. We exchange the moves, reducing the height of the line in between.

Bend-bend cancellation. The first move is a left bend that is connected to the right bend. We replace the bends by an x -monotone curve, reducing the number of bends.

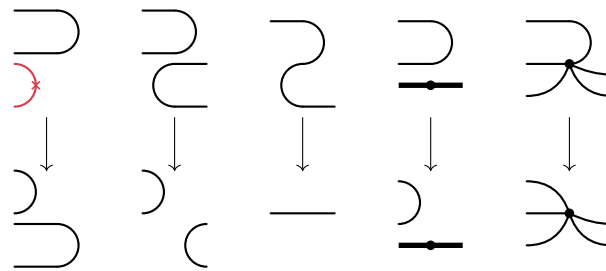
Vertex-bend cancellation. The first move is a vertex that is connected to the right bend. We replace the bend by an x -monotone curve, reducing the number of bends. We call a vertex-bend cancellation *strong* if the simplification does not decrease the absolute difference between the number of edges incident to the left and right of the vertex.

We say that a drawing ϕ is *strongly simplified* if no simplification move is possible, and *simplified* if only strong simplification moves are possible.

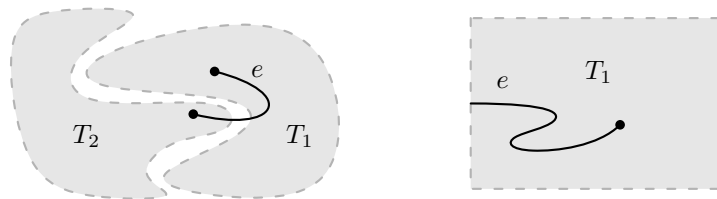
We say that ϕ is *balanced* if for any vertex v , the heights of the vertical lines immediately to the left and right of v are equal if the degree of v is even, and differ by 1 if the degree of v is odd. Balanced drawings will be useful for our algorithms. However, strong vertex-bend cancellations may make vertices less balanced.



■ **Figure 5** A right bend (marked) stuck around a vertex (left) or stuck around a bend (right). The bold line represents a bundle of arbitrarily many edges incident to the vertex.



■ **Figure 6** Left to right: stuck slide, bend-bend separation, bend-bend cancellation, vertex-bend separation, (strong) vertex-bend cancellation.



■ **Figure 7** Two local disks (containing sub-trees T_1 and T_2) with a single boundary edge e (left). An exposed drawing of the sub-tree T_1 with anchor edge e (right).

► **Lemma 2.** *If there is a drawing ϕ of height H of a local disk D , then there exists a balanced simplified drawing of D of height at most H with a bounded number of moves.*

► **Lemma 3.** *Any simplified drawing of height H of a local disk D with n vertices has at most $(H + 1)n$ moves if $n > 0$, and at most H moves if $n = 0$.*

► **Observation 4.** *Let ϕ be a balanced drawing of the local disk D . Then applying all possible non-strong simplifying moves to h keeps the drawing balanced.*

5 Bubbling a sub-tree

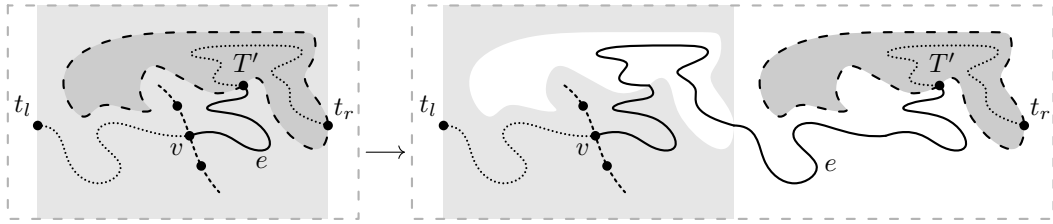
Let e be an edge of a tree T . There are two sub-trees T_1, T_2 of T that result from removing e . For $i \in \{1, 2\}$, we call the rooted trees $T_i = T_i(e)$, with the root chosen to be the endpoint of e in T_i , the *rooted sub-trees anchored via the edge e* and the edge e the *anchor edge* of the rooted tree T_i . We call the endpoint of e which is not the root of T_i the *anchor vertex* of T_i . The *exposed height* of the sub-tree $T_i \cup \{e\}$, denoted $eH(T_i, e)$, is the height of the optimal height drawing of a local disk containing T_i in its interior and such that the anchor edge e is the single boundary edge, see Figure 7. We call such a drawing of T_i an *exposed drawing* of the sub-tree T_i with respect to e .

Let P be a simple path (possibly of length¹ 0) in T . The *neighborhood* of P , denoted $N(P)$, is the subset of vertices of T not in P which are connected in T to some vertex of P by an edge. We say that a rooted sub-tree $T'' \subset T$ is a sub-tree *anchored at P* if $V(T'') \subset V(T) - V(P)$ and the root of T'' is in $N(P)$. It follows that, for any P , the edge-set of T is partitioned into three sets, the edges of P , the edges of sub-trees anchored at P , and the anchor edges incident to P .

¹ The length of a path is the number of its edges.

We call an exposed drawing of a sub-tree T' in a drawing of T a *bubble* if the strip of the plane containing this exposed drawing contains no moves of the rest of the drawing. In other words, we can compress the exposed drawing of T' into a drawing inside an arbitrary small bubble, without affecting the height of the drawing of T .

Let ϕ be a drawing of the tree T and let $T' \subset T$ be a sub-tree anchored using an edge e to a vertex v . We say that ϕ' is obtained by *bubbling the sub-tree T' at the point $t \in \mathbb{R}^2$* if ϕ' is such that i) $H(\phi') \leq H(\phi)$, ii) T' is drawn in a bubble with e the boundary edge and t being the point of e on the boundary, iii) the drawing is changed only over T' and the edge e , and iv) the ϕ' -image of e is contained in ϕ -image of $T' \cup e$. See Figure 8 for an example, where $t = t_r$. One of our main observations is that bubbling is always possible at a suitable t .



■ **Figure 8** Bubbling the sub-tree T' , t_l and t_r are locations of extreme moves in the given drawing.

► **Lemma 5.** *Let T be a tree and h be a drawing of T . Let T' a sub-tree anchored at a vertex v . Let t_l and t_r be the points of T which have the smallest and the largest x -coordinate, respectively. We can assume these points are unique. If T' contains exactly one of t_l and t_r , then T' can be bubbled at that point.*

6 Spine disks

Let D be a local disk and $B(D)$ the set of its boundary edges. Recall that the boundary edges of a local disk are divided into left boundary edges, $B_L(D)$, and right boundary edges, $B_R(D)$. Let $e_l \in B_L(D)$ and $e_r \in B_R(D)$. We say that e_l and e_r are *opposite* one another if they are incident to the same vertex in the interior of D . We call a local disk a *spine disk* with *spine* P , if all of the following hold:

1. P is a simple path in T_D , such that every boundary edge is incident to a vertex of P , and P is interior-disjoint from boundary edges of T_D .
2. There is a bijection $\alpha: B_L(D) \rightarrow B_R(D)$ such that each e is opposite $\alpha(e)$.
3. If P has at least two vertices, there are boundary edges incident to its extremal vertices.

If there are no boundary edges, we let P be an arbitrary vertex, so that P is always defined. Therefore a local disk that contains all of the input tree T and P chosen to be any vertex is a spine disk.

A *skew spine disk* is a spine disk to which a new boundary edge incident to some vertex of P is added. It follows that the height of one boundary line of any drawing of a skew spine disk is one more than the height of the other boundary line. We call a (skew) spine disk a *vertex disk* if there is a single vertex in its interior. Figure 9 shows an optimal drawing of a tree and a “decomposition” of the drawing into spine (rectangle) and skew spine disks (trapezoids). Note that a skew spine disk with a single boundary edge is a bubble. All skew spine disks in Figure 9 are bubbles.

6.1 Spine decomposition

We introduce some terminology before stating one of our main propositions. Let D be a local disk and let C be a collection of disjoint shortcuts (combinatorially distinct from the left and the right boundary lines) in D , and let C cut the disk D into disks D_1, \dots, D_m . According to Lemma 1, an optimal drawing ϕ of D can be obtained by gluing optimal drawings ϕ_i of the D_i , $i = 1, \dots, m$. Then we say ϕ is obtained by *merging* the drawings ϕ_1, \dots, ϕ_m . In our schematics, we draw a rectangle for a spine disk and a trapezoid for a skew disk. The shorter side of the trapezoid has one less boundary edge than the long side. A vertex inside a rectangle or a trapezoid indicates a vertex disk. A thick black line is a collection of parallel lines whose number is indicated. A *pipe* in a drawing bounded by two vertical lines l and r is a subpath of an edge of T drawn as an x -monotone curve between these lines. Observe that a pipe can always be drawn as a straight line connecting the lines l and r .

If D is a vertex spine disk or vertex skew spine disk then there is a trivial, straight-line, optimal drawing of D . These disks form the building blocks of our drawings. The following proposition shows how more complicated (skew) spine disks can be decomposed into less complicated ones and eventually into vertex (spine) disks.

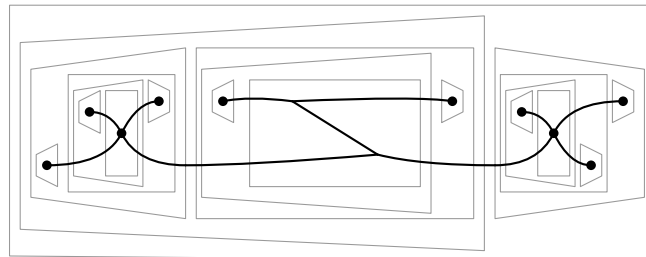
► **Proposition 6** (Spine Decomposition). *Let D be a spine (resp. skew spine) local disk with $b \geq 0$ boundary edges on one side and b (resp. $b + 1$) boundary edges on the other side. If D is not a vertex disk and D has a drawing of height H , then D has a drawing of height at most H that can be decomposed as one of the cases of Figure 10 (resp. 11), up to horizontal and vertical reflection. In these drawings m, a_i, c_j are non-negative integers.*

6.2 Structure tree

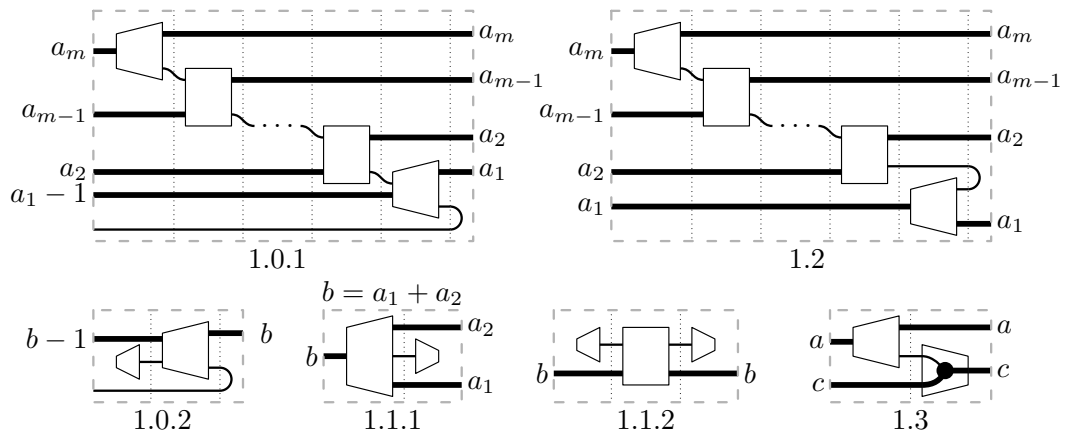
Recursive applications of Proposition 6 to an optimal-height drawing of a spine or a skew spine disk, for instance one containing all of T , result in an optimal drawing that has a hierarchical structure. Any node in the hierarchy is a spine or skew spine disk, and a node is decomposed into its children using one of the possibilities of Proposition 6. The leaves of the hierarchy are vertex disks. We call this hierarchy a *structure tree* of the optimal drawing. We call a drawing which has such a structure tree a *structured drawing*. For instance the drawings output by Proposition 6 are structured drawings.

7 Optimizing the optimal-height drawings

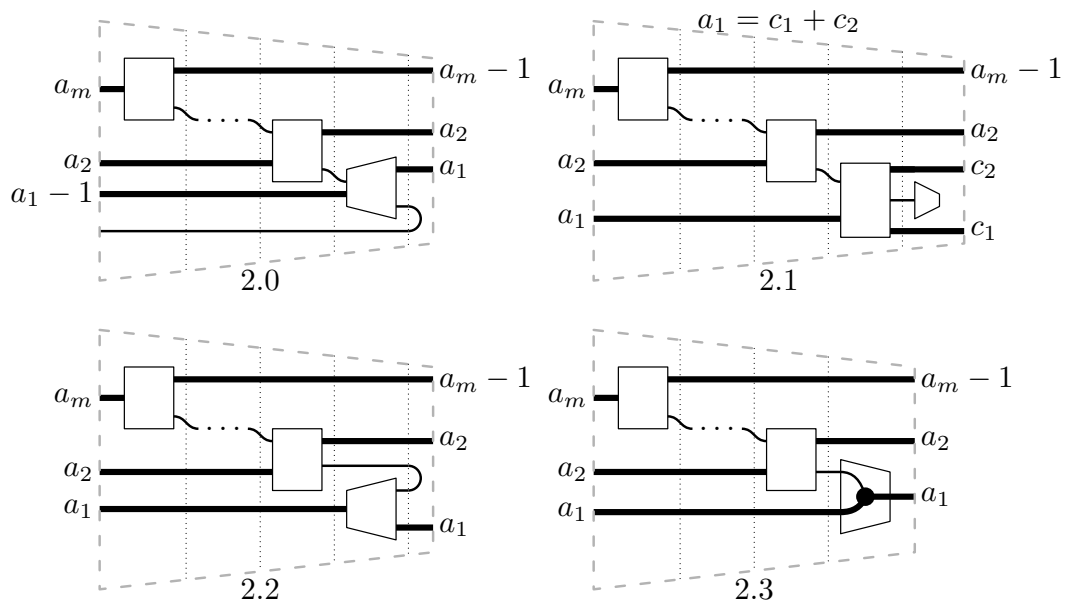
In this section, we first define the quality of drawings. We then consider those optimal drawings that maximize this quality measure.



■ **Figure 9** Spine and skew spine disks.



■ **Figure 10** Decomposition of spine disks. Thick lines indicate bundles of parallel edges. The number of parallel edges in bundles are indicated by the labels on the sides. The values a_i can be 0. Rectangles indicate spine disks and trapezoids indicate skew spine disks. A black dot indicates a vertex disk.



■ **Figure 11** Decomposition of skew spine disks.

7.1 Quality of a drawing

Let ϕ be any drawing of a local disk D . We denote by $\Lambda' = \Lambda'(\phi) = \{\lambda_i\}$ the set of combinatorially distinct vertical lines in the plane (that lie in general position with the drawing). That is, the strip S_i bounded by λ_i and λ_{i+1} , after removing pipes, is either: i) a vertex move, that is, contains a single vertex and no bends, or ii) contains a single bend and no vertices. Such a set of vertical lines can be chosen for any drawing h in general position. Consider a strip S_{ij} bounded by λ_i and λ_j . If S_{ij} contains only pipes we remove λ_i or λ_j (whichever is to the right of the other) and all the lines in between form Λ' , and repeat this operation. Let $\Lambda = \Lambda(\phi)$ be the remaining set of vertical lines. We again consider strips S_{ij} bounded by λ_i and λ_j in Λ . If after removing pipes from the strip S_{ij} it becomes a bubble, spine disk, skew spine disk, or bend, we respectively say that S_{ij} is a bubble, spine disk, skew spine disk or bend. Recall that a bubble is a special type of skew spine disk with only one boundary edge in one side.

Note that bubbles are either disjoint or nested and therefore give rise to a hierarchical structure. We say that a vertex v is of *depth* d if it is contained in exactly d bubbles. That is, there are exactly d strips, bounded by the lines of Λ , that contain the vertex and that are bubbles. We define the *depth* of a bubble and a (skew) spine disk analogously to depth of a vertex to be the number of bubbles that properly contain them.

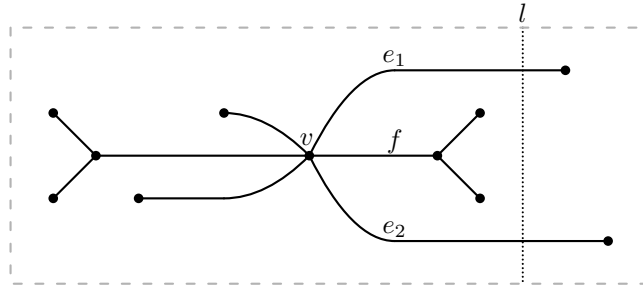
We say that a line $\lambda \in \Lambda(\phi)$ is of depth i if it is contained in exactly i strips which are bubbles, and in none of them it is a boundary. For instance, lines of depth 0 do not cut any bubble. Let $\Lambda_i = \Lambda_i(\phi)$ denote the set of lines of depth i . Let $\Lambda_{i,j} = \Lambda_{i,j}(\phi) \subset \Lambda_i$ be the set of lines of height j and depth i , and let $\delta_{i,j}(\phi) = |\Lambda_{i,j}|$ be the number of lines of depth i and height j . Note that $\delta_{i,j}(\phi)$ can be 0. Let $\Delta_i(\phi)$ be the sequence $(\delta_{i,0}(\phi), \delta_{i,1}(\phi), \dots)$, and define the *quality* of the drawing ϕ as

$$Q(\phi) = (\Delta_0(\phi), \Delta_1(\phi), \dots).$$

For two drawings ϕ and ϕ' , we compare their qualities $Q(\phi)$ and $Q(\phi')$ lexicographically, where we also compare the sequences $\Delta_i(\phi)$ to $\Delta_i(\phi')$ lexicographically. Specifically, a drawing of maximum quality maximizes the depth sequences Δ_i from left to right. That is, we are interested in the drawings where the sequence Δ_0 is maximized, and among these the sequences where Δ_1 is maximized, and so on. We emphasize that maximizing the quality does not necessarily minimize the height of the drawing. Instead, we merely use the quality measure to reduce the search space for minimum height drawings.

There remains still some arbitrariness in optimal drawings with maximum quality. For instance, a star with $2k$ leaves and a central vertex can be drawn with optimal height and with maximum quality in an exponentially many different ways, giving rise to exponentially many spine disks, by changing the order of the vertices. We get rid of these choices using the notion of secondary quality to be defined in Section 7.3.

By Lemma 2, there exists an optimal simplified and balanced drawing of any local disk D , and by Lemma 3, $(H + 1)n$ is an upper bound on the complexities of simplified drawings with height H . Therefore, the set of quality sequences of the set of all optimal, balanced and simplified drawings of a local disk is non-empty. Moreover, each quality sequence for such a drawing consists of at most $H(H + 1)n$ terms $\delta_{i,j}$, since the depth is at most the number of lines in Λ and each depth-sequence Δ_i contains at most $H = O(n)$ different height values.



■ **Figure 12** The edge f is sandwiched between e_1 and e_2 with respect to l .

7.2 Properties of drawings with maximum quality

Since the dynamic program only constructs structured drawings, we need to argue that the maximum quality drawing is structured.

► **Lemma 7.** *Let ϕ be a simplified, balanced drawing of a spine disk D that has maximum quality $Q(\phi)$ over all drawings with the same height as ϕ . Then ϕ is a structured drawing.*

Recall that for a path P the set of anchor edges, $A(P)$, is the set of edges which have exactly one endpoint on the path P . Also, the set of anchor edges of a spine disk D , $A(D)$, is the set of anchor edges of the spine path of D . Let D be a (balanced) spine disk with $2b$ boundary edges and $e \in A(D)$ be an anchor edge of D . Let $H(D)$ denote, as always, the optimal height of the disk D and $eH(T', e')$ denote the optimal exposed height of a sub-tree T' with respect to the edge e' . Also recall that the sub-tree T_e anchored by e is the sub-tree rooted at the endpoint of e which is not in the spine of the disk D . We say e is *light* (with respect to D) if the exposed height of the sub-tree T_e satisfies $eH(T_e, e) \leq H(D) - b + 1$.

The significance of light edges is that if we know a boundary edge e of D is light then given any drawing of D we can redraw the sub-tree T_e near the boundary of D in a small bubble without increasing the height, since the maximum height over the bubble would be $eH(T_e, e) + b - 1$ which would be at most $H(D)$.

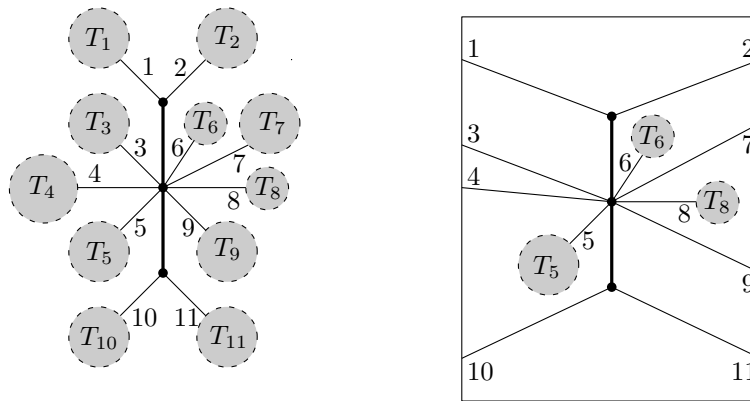
► **Lemma 8.** *Let D be a spine disk of depth d in a drawing ϕ with maximum quality $Q(\phi)$. Let $e \in A(D)$ be a light edge and T_e the sub-tree anchored by e . If e is a boundary edge then T_e is drawn in a bubble of depth d with e as the single boundary edge. Moreover, the strip between the bubble of T_e and the disk D is a sequence of bubbles of depth d anchored at the spine of D , or bends.*

7.3 Breaking ties while respecting the orders

In our arguments we will use a tie-breaking mechanism to decide between optimal drawings which all have maximum quality. We first define a perturbation of the original heights.

Let v be a vertex in D and let e_1, e_2 and f be edges incident to v . Let l be a vertical line. We say that f is *sandwiched* between e_1 and e_2 with respect to l if f does not intersect l but e_1 and e_2 intersect l , and f lies in the resulting bigon, see Figure 12. We add a small ϵ ($0 < \epsilon \ll 1$) for every sandwiched edge with respect to l to the height of l . The resulting value is called the *perturbed height* of l .

Consider the set Λ of lines of a given drawing as defined above and let $W(\phi) = (w_1, w_2, \dots)$ be the sequence of perturbed heights of lines in Λ , sorted in a non-decreasing order. Let ϕ_1 and ϕ_2 be two drawings with maximum quality. We say that ϕ_1 has a better *secondary quality* than ϕ_2 if the sequence $W(\phi_1)$ is lexicographically smaller than $W(\phi_2)$.



■ **Figure 13** Left: the path P (thick edges), anchor edges $A(P)$ numbered 1 to 11, and the anchored sub-trees. Right: A spine disk with spine path P and $b = 3$. Proposition 10 states the following. 1) Edges 5, 6, 8 are light. 2) Among the six sets $\{1\}$, $\{2\}$, $\{3, 4\}$, $\{7, 9\}$, $\{10\}$, $\{11\}$ at most one can contain a light edge. Assume it is $\{7, 9\}$. 3) If $eH(T_7) = eH(T_9) = H(D) - 3$ then $eH(T_8) \neq H(D) - 3$. If $eH(T_7) < H - 3$ and $eH(T_9) < H - 3$, then $eH(T_8) \geq H - 3$.

7.4 Fat structures

Let D be a local disk. Let ϕ be an optimal, simplified and balanced drawing such that $Q(\phi)$ is maximal among such drawings and also its secondary quality is the best possible. By Lemma 7, such a drawing has a structure tree. We call the resulting structure a *fat structure*² for D . It follows that any drawing which is optimal, simplified and balanced has to have worse or equal quality (or equal quality and equal or worse secondary quality).

The proof of the following is straightforward.

► **Lemma 9.** *Let ϕ be a drawing of local disk D with a fat structure. Then for every local disk D' , corresponding to a node in the structure tree, the restriction of the structure to D' is a fat structure.*

The following lemma allows us to enumerate the spine disks which are possible in a fat structure. Refer to Figure 13 for an example.

► **Proposition 10 (Characterization of Spine Disks in Fat Structures).** *Let P be a path in the tree and let D be a spine disk with spine path P , such that D is a node in a fat structure. Let $b > 0$ be the number of left (equivalently right) boundary edges of D .*

1. *Every edge $e \in A(P)$ that lies entirely in the interior of D is light.*
2. *All light boundary edges of D are incident to a single vertex v , and intersect the same (left or right) boundary.*
3. *For $\eta \geq 0$, let $E(\eta) \subset A(P)$ be the set of anchor edges of P , incident to v , for which the exposed height of the sub-tree anchored by that edge is $H(D) - b + 1 - \eta$. Then, for $\eta = 0$, if any edge e in $E(0)$ is not a boundary edge, then e is not sandwiched, with respect to the boundary lines, between two edges of $E(0)$ that are boundary edges. Moreover, if any edge e of $E(\geq 1) := \bigcup_{j \geq 1} E(j)$ is not a boundary edge, then e is not sandwiched between two edges of $E(\geq 1)$ that are boundary edges.*

² The name comes from the fact that the bubbles in a minimal drawing tend to contain a maximal part of the tree.

We remark that the secondary quality is needed only in the proof of the second part of statement 3. That is, the rest of proposition holds for drawings with maximum quality.

8 The dynamic program

We describe the algorithm for computing the optimal height of an input drawing. Modifying the dynamic program to compute an actual optimal height drawing is standard.

We think of row m of the dynamic programming table as containing (the description) of those spine and skew spine disks that have exactly m vertices in their interior and satisfy Proposition 10, together with their optimal heights. For $m = 1$, i.e. the first row, we must consider the (skew) spine disks with exactly one vertex in their interior. Since we are interested in balanced drawings, we know that each vertex v of even degree defines $O(d(v))$ distinct spine disks, where $d(v)$ is the degree of the vertex v . These are given by all the $O(d(v))$ possible balanced partitions of the edges incident to v into left and right edges, maintaining the order around v . The optimal drawings are trivial. Similarly, vertices with odd degree determine $O(d(v))$ distinct skew spine disks.

Assume that we have populated the table up to row $m - 1$. The algorithm first computes all spine and skew spine disks with m vertices that satisfy Proposition 10. If $m = n$ we take the spine disk containing all the tree. If we are computing skew spine disks we take all the bubbles with m vertices (there are at most $2(n - 1)$ bubbles). The rest of the (skew) spine disks are computed as follows. We determine only spine disks, and this also determines all possible skew spine disks. This is because a skew spine disk is the result of changing one non-boundary anchor edge of a spine disk into a boundary edge.

If we know the exposed heights of anchored sub-trees, then Proposition 10 implies that a spine disk is determined uniquely given the following parameters. We also indicate an upper bound on the number of possibilities for each of them.

1. The spine path P : $O(n^2)$ possibilities.
2. The boundary height b : $O(n)$ possibilities.
3. The height H : $O(n)$ possibilities.
4. A partition of $A(P)$ into cyclically contiguous subsequences $A_L(P)$ and $A_R(P)$: $O(n^2)$ possibilities.
5. The vertex v to which light boundary edges are incident: $O(n)$ possibilities.
6. Two consecutive sequences of edges around v : one for $E(0)$ and the other for $E(\geq 1)$: $O(n^4)$ possibilities.

Therefore, there are polynomially many possible values for all these parameters, namely $O(n^{11})$. A particular set of values may or may not define a valid spine disk, and a more careful analysis may result in asymptotically fewer relevant values. It is straightforward to compute the disk (if any) that corresponds to a particular set of values for the parameters.

Let p be the number of vertices in (some choice for) the spine path P . If the sub-tree anchored by an edge e has more than $m - p$ vertices then e has to be a boundary edge. Otherwise, the exposed height of the anchored sub-tree can be read from the table since it has at most $m - 1$ vertices and is a bubble. Thus, we can determine which set of parameters determine a spine disk with m internal vertices. Additional details can be found in the full version of the paper.

After determining possible (skew) spine disks, we compute their optimal heights. This can be done by considering the polynomially many different ways that a (skew) spine disk can be decomposed into (skew) spine disks of smaller complexity (i.e vertices and edges), given in Figures 10 and 11. For a given (skew) spine disk D , the number of possibilities is

determined by the number of its possible first and last moves. These moves are either bends or vertices which are determined by an edge or a vertex of the tree, respectively. It follows that the total number of possibilities is polynomial. We consider all of the polynomially many decompositions of D . For a given decomposition, we read the optimal heights of their inner disks from the table. These heights can be used to derive the height of a drawing of D . The optimal height of D is the minimum such value over all of its decompositions. For more details to this part of the algorithm, and an exception to the general description above, we refer to the full version of the paper, where we prove the following.

► **Theorem 11.** *Let D be a (skew) spine disk. There is a polynomial-time algorithm for drawing D with optimal height.*

9 Discussion

We have presented the first polynomial-time algorithm for drawing plane trees with optimal height. The case of weighted plane trees remains open. Moreover, the setting of unweighted graphs remains open, but is believed to be NP-hard by some. However, we believe that a polynomial time algorithm may exist even in this setting.

If the graph setting turns out to be NP-hard, then the situation resembles that of the (non-embedded) min-cut linear arrangement problem, which has a polynomial time algorithm for unweighted trees [15], but is NP-hard for graphs [9, 8].

There are other interesting problems around the complexity and properties of optimal height drawings that might help in finding faster algorithms. As one such property, we conjecture that for unweighted trees there always exists an optimal drawing without spiraling edges. A spiral on an edge is depicted in Figure 2.

References

- 1 Hugo A Akitaya, Maarten Löffler, and Irene Parada. How to fit a tree in a box. In *Proceedings of the 26th International Symposium on Graph Drawing and Network Visualization (GD 2018)*, pages 361–367. Springer, 2018.
- 2 Therese Biedl, Erin Wolf Chambers, David Eppstein, Arnaud de Mesmay, and Tim Ophelders. Homotopy height, grid-major height and graph-drawing height. In *Proceedings of the 27th Graph Drawing and Network Visualization (GD 2019)*, pages 468–481. Springer, 2019.
- 3 Benjamin Burton, Erin Chambers, Marc van Kreveld, Wouter Meulemans, Tim Ophelders, and Bettina Speckmann. Computing optimal homotopies over a spiked plane with polygonal boundary. In *Proceedings of the 25th Annual European Symposium on Algorithms (ESA)*, 2017.
- 4 Erin Wolf Chambers, Gregory R Chambers, Arnaud de Mesmay, Tim Ophelders, and Regina Rotman. Constructing monotone homotopies and sweepouts. *arXiv preprint*, 2017. [arXiv: 1704.06175](https://arxiv.org/abs/1704.06175).
- 5 Erin Wolf Chambers, Arnaud de Mesmay, and Tim Ophelders. On the complexity of optimal homotopies. In *Proceedings of the 29th Annual Symposium on Discrete Algorithms (SODA 2018)*, pages 1121–1134, 2018.
- 6 Erin Wolf Chambers and David Letscher. On the height of a homotopy. In *Canadian Conference on Computational Geometry (CCCG)*, volume 9, pages 103–106, 2009.
- 7 Moon-Jung Chung, Fillia Makedon, Ivan Hal Sudborough, and Jonathan Turner. Polynomial time algorithms for the min cut problem on degree restricted trees. *SIAM Journal on Computing*, 14(1):158–177, 1985.
- 8 Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. Freeman San Francisco, 1979.

55:16 Minimum Height Drawings of Ordered Trees in Polynomial Time

- 9 F Gavril. Some NP-complete problems on graphs. In *Proc. Conf. on Inform. Sci. and Systems, 1977*, pages 91–95, 1977.
- 10 Sarel Har-Peled, Amir Nayyeri, Mohammad Salavatipour, and Anastasios Sidiropoulos. How to walk your dog in the mountains with no magic leash. *Discrete & Computational Geometry*, 55(1):39–73, 2016.
- 11 Thomas Lengauer. Upper and lower bounds on the complexity of the min-cut linear arrangement problem on trees. *SIAM Journal on Algebraic Discrete Methods*, 3(1):99–113, 1982.
- 12 Debajyoti Mondal, Muhammad Jawaherul Alam, and Md. Saidur Rahman. Minimum-layer drawings of trees. In Naoki Katoh and Amit Kumar, editors, *WALCOM: Algorithms and Computation*, pages 221–232. Springer, 2011.
- 13 B. Monien and I.H. Sudborough. Min cut is NP-complete for edge weighted trees. *Theoretical Computer Science*, 58(1):209–229, 1988.
- 14 Yossi Shiloach. A minimum linear arrangement algorithm for undirected trees. *SIAM Journal on Computing*, 8(1):15–32, 1979.
- 15 Mihalis Yannakakis. A polynomial algorithm for the min-cut linear arrangement of trees. *Journal of the ACM*, 32(4):950–988, 1985.

Disjointness Graphs of Short Polygonal Chains

János Pach ✉

Rényi Institute, Budapest, Hungary
MIPT, Moscow, Russia

Gábor Tardos ✉

Rényi Institute, Budapest, Hungary
MIPT, Moscow, Russia

Géza Tóth ✉

Rényi Institute, Budapest, Hungary

Abstract

The *disjointness graph* of a set system is a graph whose vertices are the sets, two being connected by an edge if and only if they are disjoint. It is known that the disjointness graph G of any system of segments in the plane is χ -bounded, that is, its chromatic number $\chi(G)$ is upper bounded by a function of its clique number $\omega(G)$.

Here we show that this statement does not remain true for systems of polygonal chains of length 2. We also construct systems of polygonal chains of length 3 such that their disjointness graphs have arbitrarily large girth and chromatic number. In the opposite direction, we show that the class of disjointness graphs of (possibly self-intersecting) *2-way infinite* polygonal chains of length 3 is χ -bounded: for every such graph G , we have $\chi(G) \leq (\omega(G))^3 + \omega(G)$.

2012 ACM Subject Classification Mathematics of computing → Graph coloring

Keywords and phrases χ -bounded, disjointness graph

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.56

Funding *János Pach*: Supported by the National Research, Development and Innovation Office (NKFIH) grant K-131529, ERC Advanced Grant “GeoScape,” the Austrian Science Fund grant Z 342-N31, and by the Ministry of Education and Science of the Russian Federation in the framework of MegaGrant No. 075-15-2019-1926.

Gábor Tardos: Supported by the ERC Synergy Grant “Dynasnet” No. 810115, the ERC advanced grant “GeoSpace” No. 882971, the National Research, Development and Innovation Office – NKFIH projects K-132696 and SSN-135643.

Géza Tóth: Supported by National Research, Development and Innovation Office, NKFIH, K-131529 and ERC Advanced Grant “GeoScape.”

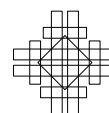
1 Introduction

Ramsey theory has many applications to other parts of mathematics and computer science [27], including complexity theory [21], approximation algorithms, [22], coding [18], geometric data structures [20], graph drawing and representation [2]. Constructing nearly optimal Ramsey graphs is a notoriously difficult combinatorial problem [10]. The few efficient constructions that we have are far from optimal, but they can come in handy in those areas where we have interesting theorems, but lack nontrivial constructions. Here we provide two examples from combinatorial geometry, based on two classical constructions of Erdős and Hajnal [9, 8]. We close this paper with a result pointing in the opposite direction.



© János Pach, Gábor Tardos, and Géza Tóth;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 56; pp. 56:1–56:12
Leibniz International Proceedings in Informatics
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



For any graph G , let $\chi(G)$ and $\omega(G)$ denote the *chromatic number* and the *clique number* of G , respectively. Clearly, we have $\chi(G) \geq \omega(G)$, and if equality holds for every induced subgraph of G , then G is called a *perfect graph*. Following Gyárfás and Lehel [15, 16, 13, 14], a class of graphs \mathcal{G} is said to be χ -*bounded* if there is a function f such that $\chi(G) \leq f(\omega(G))$ for every $G \in \mathcal{G}$.

Which classes of graphs are χ -bounded? Or, reversing the question, if a graph has small clique number, how can its chromatic number be large? These questions are related to the some of the deepest unsolved problems in graph theory. There are two different approaches that have yielded spectacular results in recent years.

One can investigate what kind of substructures must necessarily occur in graphs of high chromatic number. According to Hadwiger's conjecture [17], if the chromatic number of a graph is at least t , then it must contain a K_t -minor. (We now know that it contains a K_s -minor with $s = \Omega(t/(\log \log t))$; cf. [7].) Gyárfás [12] proved that if a graph has bounded clique number and its chromatic number is sufficiently large, then it must contain a long induced path; see also [11]. According to the (still open) Gyárfás-Sumner conjecture [29], the same is true for any fixed tree instead of a path. Scott and Seymour proved that the class of graphs with no induced odd cycle of length at least 5 is χ -bounded. For many beautiful recent results of this kind, see the survey [28].

The second fruitful research direction was initiated by Asplund and Grünbaum [1]: Find geometrically defined classes of graphs that are χ -bounded. Given a set S of geometric objects, their *intersection graph* (resp., *disjointness graph*) is a graph on the vertex set S , in which two vertices are connected by an edge if and only if the corresponding objects have a nonempty intersection (resp., are disjoint). It was proved in [1] that the class of intersection graphs of axis-parallel rectangles in the plane is χ -bounded (see also [4]). The corresponding statement is false for boxes in 3 and higher dimensions [3], and even for segments in the plane [26].

For *disjointness* graphs G of systems of segments in the plane, we have $\chi(G) \leq (\omega(G))^4$ [19]. The same is true for systems *x-monotone curves*, that is, for continuous curves in the plane with the property that every vertical line intersects them in at most one point. It was shown in [25] that, in this generality, the order of magnitude of this bound cannot be improved. On the other hand, we proved [24] that the class of disjointness graphs of *strings* (continuous curves in the plane) is not χ -bounded. Improving our construction, Mütze, Walczak, and Wiechert [23] exhibited systems of polygonal curves consisting of *three* segments such that their disjointness graphs are triangle-free ($\omega = 2$), yet their chromatic numbers can be arbitrarily large.

The above results leave open the case of polygonal curves consisting of *two* segments. Our first result settles this case. A polygonal curve consisting of k segments is called a *polygonal k-chain*.

► **Theorem 1.** *There exist arrangements of polygonal 2-chains in the plane whose disjointness graphs are triangle-free and have arbitrarily large chromatic numbers.*

We do not know if Theorem 1 can be strengthened by requiring that the disjointness graph of the curves has large girth.

► **Problem 2.** *Do there exist arrangements of polygonal 2-chains in the plane whose disjointness graphs have arbitrarily large girth and chromatic number?*

Our next result shows that the answer to the above question is in the affirmative if, instead of 2-chains, we are allowed to use polygonal 3-chains.

► **Theorem 3.** *For any integers g and k , there is an arrangement of non-selfintersecting polygonal 3-chains in the plane whose disjointness graph has girth at least g and chromatic number at least k .*

A 1-way infinite polygonal 2-chain is the union of a half-line and a segment that share an endpoint. In our proof of Theorem 1, we actually construct arrangements of 1-way infinite polygonal 2-chains whose disjointness graphs are triangle free, but have arbitrarily large chromatic number. Doubly tracing these 1-way infinite 2-chains and slightly perturbing the resulting curve, we obtain an arrangement of 2-way infinite 4-chains, i.e., 4-chains whose first and last pieces are half-lines. Hence, we obtain the following

► **Corollary 4.** *There exist arrangements of 2-way infinite polygonal 4-chains in the plane whose disjointness graphs are triangle-free and have arbitrarily large chromatic numbers.*

Our next theorem shows that Corollary 4 is optimal: the class of disjointness graphs of (possibly self-intersecting) 2-way infinite polygonal 3-chains is χ -bounded.

► **Theorem 5.** *Let G be the disjointness graph of an arrangement of 2-way infinite polygonal 3-chains in the plane. Then we have $\chi(G) \leq (\omega(G))^3 + \omega(G)$.*

In fact, we will establish Theorem 5 in a somewhat stronger setting: for arrangements of 2-way infinite curves that consist of three x -monotone pieces; see Theorem 7. With more work, the bound in Theorem 5 and Theorem 7 can be improved to $\chi(G) \leq (\omega(G))^3$.

In the polygonal case, our proof is algorithmic. There is a polynomial time algorithm in the number of the polygonal chains, which, for every k , either finds k pairwise disjoint chains or produces a coloring of their disjointness graph with at most k^3 colors.

In Sections 2 and 3, we establish Theorems 1 and 3, respectively. Section 4 contains the proof of Theorem 5. We end this note with a few remarks and open problems.

In what follows, we informally call a polygonal 2-chain a *V-shape* and a polygonal 3-chain a *Z-shape*.

2 Shift graphs – Proof of Theorem 1

For every $n > 1$, Erdős and Hajnal [8] defined the *shift graph* S_n , as follows. The vertex set of S_n consist of all pairs (a, b) with $1 \leq a < b \leq n$, where two vertices, (a, b) and (a', b') , are connected by an edge if and only if $b = a'$ or $b' = a$. It is easy to see that S_n is triangle-free and that $\chi(S_n) = \lceil \log_2 n \rceil$.

Order the vertices (a, b) of S_n according to the *co-lexicographic order*, that is, let $(a, b) \prec (a', b')$ if $b < b'$, or if $b = b'$ and $a < a'$. Let $v_1, \dots, v_{\binom{n}{2}}$ denote the vertices of S_n , listed in this order.

Let $v_i = (a, b)$ be a vertex. Its neighbors having a smaller index are (a', b') with $b' = a$. No such neighbor exist if and only if $a = 1$. Notice that, for any i ,

1. either v_i has no neighbor v_j with a smaller index $j < i$,
2. or there exist integers $c(i), d(i)$ with $1 \leq c(i) \leq d(i) < i$ such that for every $j < i$,

$$v_j v_i \in E(S_n) \iff c(i) \leq j \leq d(i).$$

Recall that a 1-way infinite *V-shape* is the union of a half-line and a segment that share an endpoint. In the rest of this proof, for simplicity, we call a 1-way infinite V-shape *long*.

Our goal is to assign a long V-shape to each vertex of S_n so that two V-shapes are disjoint if and only if the corresponding vertices are adjacent in S_n . This will prove Theorem 1, because in any finite collection of long V-shapes, we can cut the half-lines short so that

the resulting (bounded) V-shapes have the same intersection structure. Hence, we obtain a collection of V-shapes with S_n as its disjointness graph, and the graphs S_n are triangle-free and their chromatic numbers tend to infinity, as $n \rightarrow \infty$.

We assign the long V-shape V_i to the vertex v_i of S_n recursively starting at V_1 . Let h_i and s_i denote the half-line and the straight-line segment, respectively, comprising V_i and let us denote their common endpoint by $p_i = (x_i, y_i)$. We write q_i for the other endpoint of s_i .

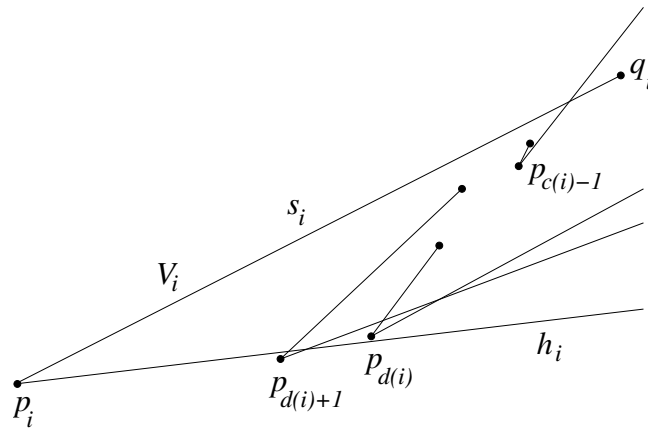
During the recursive process, we will maintain the following properties:

- (i) p_i is the left end point of both h_i and s_i ;
- (ii) both h_i and s_i have positive slopes;
- (iii) s_i is above h_i , i.e., the slope of s_i is larger than the slope of h_i ;
- (iv) for any $i > j$, the slope of h_i will be smaller than the slope of h_j ;
- (v) for any $i > j$, we have $x_i < x_j$ and $y_i < y_j$.

Let V_1 be any long V-shape satisfying the above conditions. Let $i > 1$, and assume recursively that we have already constructed the long V-shapes V_1, \dots, V_{i-1} satisfying the above requirements. Next, we define V_i . We distinguish two cases:

Case A: The vertex $v_i = (a, b)$ has no neighbor with a smaller index, i.e., we have $a = 1$.

Let ℓ be a horizontal line passing above p_1 . It will intersect every V_j with $1 \leq j < i$. Slightly rotate ℓ about any fixed point of the plane so that the resulting line ℓ' has a very small positive slope, smaller than the slope of h_{i-1} and it still intersects all V_j for $j < i$. Choose a point $p_i = (x_i, y_i) \in \ell'$, very far to the left, so that $x_i < x_{i-1}$ and $y_i < y_{i-1}$. Let h_i be the part of ℓ' to the right of p_i , and let q_i be a point to the right of p_i which lies above h_i . One can choose q_i such that the segment $s_i = p_i q_i$ does not intersect any of the earlier V_j .



■ **Figure 1** Inserting V_i .

Case B: The vertex $v_i = (a, b)$ has at least one neighbor of smaller index, i.e., $a > 1$.

Let $c(i)$ and $d(i)$ be the constants satisfying property (2) above and let ℓ be a horizontal line that passes below $p_{d(i)}$ and above $p_{d(i)+1}$. In case $d(i) + 1 = i$ we could simply choose ℓ to be an arbitrary horizontal line below $p_{d(i)}$, but the careful reader may notice that this case never occurs as no vertex v_i in S_n is adjacent to v_{i-1} .

The line ℓ intersects every V_j with $d(i) < j < i$ and is disjoint from all V_j with $j \leq d(i)$. Slightly rotate ℓ about any fixed point in the plane so that the resulting line ℓ' has a very small positive slope, smaller than that of h_{i-1} and it still intersects the same previously defined long V-shapes V_j . Select a slope α which is larger than the slope of $h_{c(i)}$, but smaller than the slope of $h_{c(i)-1}$, if $h_{c(i)-1}$ exists, that is, if $c(i) > 1$.

For any $j < i$, let ℓ_j and ℓ'_j denote the lines of slope α through p_j and q_j , respectively. Choose a point $p_i = (x_i, y_i) \in \ell'$ so far to the left that we have $x_i < x_{i-1}$, $y_i < y_{i-1}$ and p_i lies above the lines ℓ_j and ℓ'_j , for all $j \leq i$.

Let h_i be the part of ℓ' to the right of p_i . Let f be the half-line of slope α , whose left endpoint is p_i . Then f goes strictly above all s_j for $j < i$ and also of all h_j with $c(i) \leq j < i$, but will intersect all h_j with $1 \leq j < c(i)$. Choose q_i on f to the right of these intersection points, then the segment $s_i = p_i q_i$ also intersects all h_j with $1 \leq j < c(i)$.

Notice that the long V-shape V_i consisting of h_i and s_i constructed above satisfies the conditions (i)–(v) listed above, further it intersects exactly those other long V-shapes V_j ($j < i$) for which v_j and v_i are not adjacent in S_n . See Fig. 1. This means that the disjointness graph of the collection of the $\binom{n}{2}$ long V-shapes constructed above is exactly S_n . This completes the proof of Theorem 1. ◀

In the above proof, we have constructed a collection of 1-way infinite V-shapes in which each pair intersects at most twice. With a little additional care (namely, by insisting that each q_i is higher than p_1), we can achieve the following. For $1 \leq i < j \leq \binom{n}{2}$, with $v_i = (a, b)$ and $v_j = (a', b')$, we have

- if $a' < b$, then V_i and V_j intersect once;
- if $a' = b$, then V_i and V_j are disjoint;
- if $a' > b$, then V_i and V_j intersect twice.

3 Hypergraphs of large girth – Proof of Theorem 3

A hypergraph H is a pair (V, E) , where V is a finite vertex set, E is the set of hyperedges, that is, a collection of subsets of V . It is called n -uniform if each of its hyperedges has n vertices. In a *proper coloring* of H , every vertex is assigned a color in such a way that none of the hyperedges is monochromatic. The *chromatic number* of H is the smallest number of colors used in a proper coloring of H . A *Berge-cycle* in H consists of a sequence of distinct vertices v_1, \dots, v_k and a sequence of distinct hyperedges $e_1, \dots, e_k \in E$ with $v_i, v_{i+1} \in e_i$ for $1 \leq i < k$ and $v_k, v_1 \in e_k$. Here k is the *length* of the Berge-cycle and it is assumed to be at least 2. The *girth* of a hypergraph is the length of its shortest Berge-cycle (or infinite if it has no Berge-cycle).

For the proof, we need the following classical result.

► **Erdős-Hajnal Theorem** ([9], Corollary 13.4). *For any integers $n \geq 2$, $g \geq 3$, and $k \geq 2$, there exists an n -uniform hypergraph with girth at least g and chromatic number at least k .*

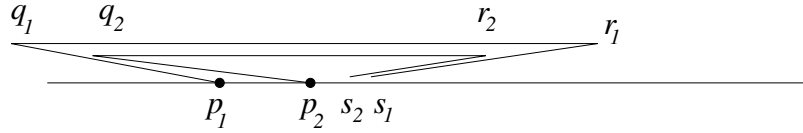
Theorem 3 is a direct consequence of part (5) of the following statement.

► **Lemma 6.** *For any integers $g \geq 3$, $k \geq 2$, there is a natural number $n = n(g, k)$ such that for every set P of n points on the x -axis in \mathbf{R}^2 and for every real $c > 0$, there is an arrangement $Z = Z(P)$ of n Z-shapes satisfying the following conditions.*

- (1) *Each point in P is the endpoint of exactly one Z-shape in Z .*
- (2) *Apart from a single endpoint in P , every Z-shape in Z lies strictly above the x -axis.*
- (3) *No Z-shape in Z is self-intersecting and any two cross at most twice.*
- (4) *For any Z-shape $z = pqr \in Z$ whose vertices p, q, r, s have x -coordinates x_p, x_q, x_r, x_s , and $p \in P$, we have $x_q + c < x_p < x_s < x_r - c$.*
- (5) *The disjointness graph of the Z-shapes in Z has girth at least g and chromatic number at least k .*

56:6 Disjointness Graphs of Short Polygonal Chains

Proof. For each g , we prove the lemma by induction on k . We fix $g \geq 3$. For $k = 2$, $n(g, 2) = 2$ is a good choice. For any two points on the x -axis and any $c > 0$, we can take two disjoint Z-shapes satisfying the requirements. Their disjointness graph is K_2 , its chromatic number 2 and it has infinite girth. See Fig. 2.



■ **Figure 2** The case $k = 2$.

Suppose now that $k \geq 2$ and that we have already proved the statement for k . Now we prove it for $k + 1$. Let $n = n(g, k)$.

By the Erdős-Hajnal Theorem stated above, there exists an n -uniform hypergraph H whose girth is at least g and chromatic number at least $k + 1$. Let v_1, v_2, \dots, v_m denote the vertices of H and e_1, e_2, \dots, e_M the hyperedges of H . Let $N = nM + m$. We show that $n(g, k + 1) = N$ satisfies the requirements of the lemma.

Let P be an arbitrary set of N points on the x -axis and let $c > 0$. For any $v_i \in V(H)$, let d_i denote the *degree* of v_i , that is, the number of hyperedges that contain v_i . Obviously, we have

$$\sum_{i=1}^m (d_i + 1) = nM + m = N.$$

Choose m disjoint open intervals, I_1, \dots, I_m , such that each I_i contains precisely $d_i + 1$ points of P . For every i , $1 \leq i \leq m$, we associate the interval I_i with vertex v_i of H . Let p_i denote the leftmost point in $P \cap I_i$. For every i and j ($1 \leq i \leq m$, $1 \leq j \leq M$) for which $v_i \in e_j$, assign a distinct point $p_i^j \in (P \cap I_i) \setminus \{p_i\}$ to the pair (v_i, e_j) .

Next, we construct a set of N Z-shapes that satisfy conditions (1)–(5) of the lemma with parameters $g, k + 1$, and c . We construct subsets Z_j of our eventual set of Z-shapes for $1 \leq j \leq M$. We construct these sets one by one starting at Z_1 and using the inductive hypothesis for various subsets of P of size n and with a parameter c' that we choose to be larger than c plus the diameter of P .

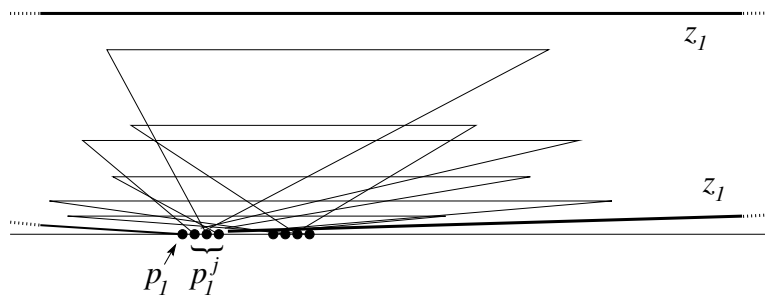
For $j = 1$, consider the $n = n(g, k)$ -element point set $P'_1 = \{p_i^1 : v_i \in e_1\}$. By the induction hypothesis, there is a set Z_1 of Z-shapes such that one of their endpoints belongs to P'_1 , and they satisfy conditions (1)–(5) with parameter c' .

Suppose that $j > 1$ and that we have already constructed the sets of Z-shapes Z_1, \dots, Z_{j-1} . Now let $P'_j = \{p_i^j : v_i \in e_j\}$. By the induction hypothesis, there is a set Z'_j of Z-shapes with one of their endpoints in P' which satisfy conditions (1)–(5) with parameter c' . Apply an affine transformation $(x, y) \rightarrow (x, y/K_j)$ to all Z-shapes in Z'_j , where K_j is a very large constant to be specified later. The resulting family of Z-shapes, Z_j , still satisfies all defining conditions and, by choosing K_j large enough, we can achieve that every element of Z_j intersects every Z-shape in $\bigcup_{h < j} Z_h$ exactly once or twice.

The set $\bigcup_{j=1}^M Z_j$ contains exactly one Z-shape starting at each point p_i^j . We still need to add one Z-shape $z_i = p_i q_i r_i s_i$ starting at each point p_i , $1 \leq i \leq m$. We define them recursively for $i = 1, \dots, m$. We make sure that each $z_i = p_i q_i r_i s_i$ satisfies the following properties.

- (i) The segment $q_i r_i$ is horizontal and the y -coordinate of its points is larger than the y -coordinate of any point of any Z -shape in $(\bigcup_{j=1}^M Z_j) \cup \{z_h : 1 \leq h < i\}$.
- (ii) The slope of $p_i q_i$ is $-\varepsilon_i$, the slope of $r_i s_i$ is ε_i , for a sufficiently small constant $\varepsilon_i > 0$, to be specified later.
- (iii) The x -coordinate of s_i is equal to the x -coordinate of the right endpoint of I_i , and the y -coordinate of s_i is ε_i .

Clearly, if we choose $\varepsilon_i > 0$ sufficiently small, then z_i is disjoint from all Z -shapes in $\bigcup_{j=1}^M Z_j$ that start in I_i , but it intersects exactly once all other Z -shapes already defined. Also, z_i satisfies conditions (2) and (3), and it satisfies condition (4), too, provided that ε_i is sufficiently small. See Fig. 3.



■ **Figure 3** Inserting z_1 .

As we maintained conditions (1)–(4) throughout the construction, it remains only to prove that the disjointness graph G of Z satisfies condition (5) with $k + 1$ in place of k .

To this end, let us explore the structure of G . The vertices of G can be partitioned into the sets Z_j for $1 \leq j \leq M$ and the independent set $W = \{z_i : 1 \leq i \leq m\}$. Further, there is no edge between two distinct sets Z_j and $Z_{j'}$. There is a single edge from z_i to Z_j if $v_i \in e_j$, and there is no edge from z_i to Z_j otherwise. Finally, each vertex in Z_j is adjacent to exactly one of the vertices z_i , and it satisfies $v_i \in e_j$.

The structure above implies that each cycle C of G is either contained in a single set Z_j , or it passes through several sets Z_j and several vertices in W . In the former case, by our assumption on the disjointness graph of Z_j , the length of C is at most g . In the latter case, let us record the vertices of W and the sets Z_j as the cycle passes through them: $z_{i_1}, Z_{j_1}, z_{i_2}, Z_{j_2}, \dots, z_{i_h}, Z_{j_h}$. Here, the vertices v_{i_1}, \dots, v_{i_h} are all distinct and, if the same is true for the hyperedges e_{j_1}, \dots, e_{j_h} , then they form a Berge-cycle of length h in the hypergraph H . If the hyperedges are not all distinct, then an even shorter Berge-cycle is formed by any repetition-free interval between two occurrences of the same hyperedge. By our assumption on the girth of H , we have $h \geq g$ in both cases, so all cycles of G have length at least g , as required.

Suppose now that there is a proper k -coloring of G . Restricting it to the set W (and identifying each $z_i \in W$ with the vertex v_i of H), we obtain a k -coloring of the vertices of the hypergraph H . By our assumption, this cannot be a proper coloring. Therefore, there is a monochromatic hyperedge e_j . In this case, no vertex in Z_j can receive the common color of the vertices of e_j , so we have a proper $(k - 1)$ -coloring of Z_j . This contradicts our assumption on the disjointness graph of Z_j and, thus, proves that G has no proper k -coloring. This concludes the proof of Lemma 6 and, hence, of Theorem 3. ◀

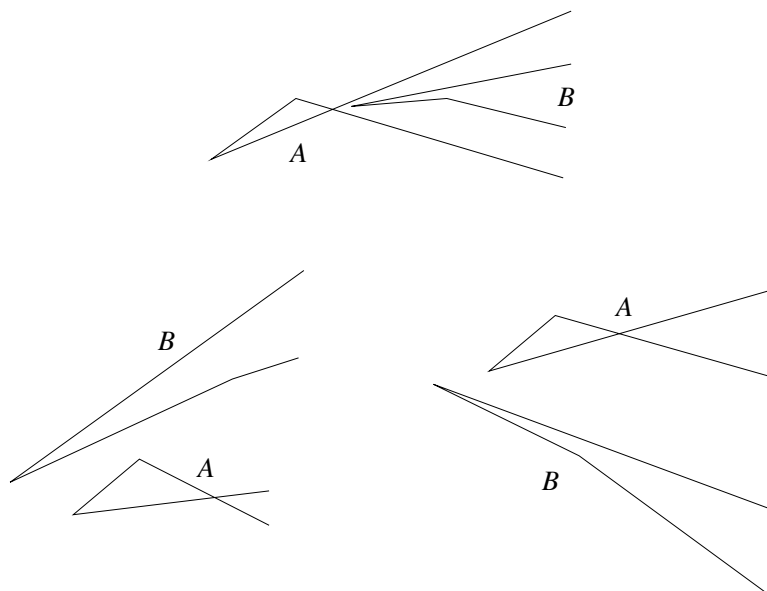
James Davies [5] used a very similar construction to show that there are intersection graphs of axis-parallel boxes and intersection graphs of lines in 3-space with arbitrarily large girths and chromatic numbers.

4 Two-way infinite polygonal chains – Proof of Theorem 5

As we pointed out at the end of Section 2, the class of disjointness graphs of 1-way infinite V-shapes is not χ -bounded. But if we require both ends of a V-shape to be long, the situation will change.

A *2-way infinite polygonal k -chain* is a continuous curve in the plane consisting of two half-lines connected by an (ordinary) polygonal $(k - 2)$ -chain. We can relax this definition by requiring only that each of the k pieces are x -monotone, and the first and the last pieces have unbounded projections to the x -axis. In this case, the curve is called a *2-way infinite k -monotone chain*.

According to this definition, a 2-way infinite polygonal 2-chain (V-shape) whose half-lines are not vertical is a 2-way infinite 2-monotone chain. It can also be regarded as a degenerate 2-way infinite 3-monotone chain. Note that by performing a suitable rotation, if necessary, we can always assume that none of the half-line pieces of a finite arrangement of 2-way infinite polygonal k -chains is vertical. Therefore, the following theorem implies Theorem 5.



■ **Figure 4** The three partial orders: A is to the left of B , below B , and above B .

► **Theorem 7.** *The disjointness graph G of a finite arrangement of 2-way infinite 3-monotone chains satisfies $\chi(G) \leq (\omega(G))^3 + \omega(G)$.*

Proof. We call a (possibly self-intersecting) 2-way infinite k -monotone chain A *wide* if it intersects every vertical line. A chain A with this property divides the plane into (open) connected components, exactly one of which contains a vertical half-line pointing upwards. We call this component the *upside of A* . For any two wide 2-way infinite k -monotone chains, A and B , we say that A is *higher* than B if A is contained in the upside of A . In this case, the upside of B is also contained in the upside of A . Therefore, the relation “higher” defines a partial order on any arrangement of wide k -monotone chains. According to this

partial order, only *disjoint* pairs are comparable. Since any two disjoint *wide* 2-way infinite k -monotone chains are comparable, the disjointness graph of any collection of wide 2-way infinite k -monotone chains is a comparability graph.

Now we turn our attention to the non-wide case. The complement of a non-wide 2-way infinite k -monotone chain A has precisely one connected component which contains a vertical line. We call this component the *large component*. The chain A is said to be a *right chain* if A is to the right of the vertical lines in the large component, otherwise it is a *left chain*. If A is a right chain, we call its large component the *left side* of A . On the other hand, if A is a left chain, we call the union of *all* connected components of the complement of A , other than its large component, the *left side* of A .

For any two non-wide 2-way infinite k -monotone chains, A and B , we say that A is *to the left of* B if both A and its left side are contained in the left side of B . Obviously, this relation also defines a partial order, with respect to which only disjoint non-wide chains are comparable. It is not true that any two disjoint non-wide 2-way infinite 3-monotone chains are comparable. Therefore, we need to introduce two further partial orders.

For any two subsets of the plane, A and B , we say that A is *below* B (A is *above* B , resp.), if the following two conditions are satisfied:

1. every vertical line that intersects A also intersects B ;
2. if $a \in A \cap \ell$ and $b \in B \cap \ell$ for a vertical line ℓ , then the y -coordinate of a is strictly *lower* (*higher*, resp.) than the y -coordinate of b .

Note that “above” and “below” are two separate partial orders and not the inverses of each other. It is clear that both of these relations are partial orders on arbitrary planar sets and that any two comparable sets are disjoint. See Fig. 4.

► **Lemma 8.** *Any two disjoint non-wide 2-way infinite 3-monotone chains, A and B , are comparable by one of the three relations “below”, “above”, or “to the left”.*

To establish the lemma, note that non-wide 2-way infinite 3-monotone chains must be, in fact, 2-way infinite 2-monotone chains. A left chain with this property is the union of the graphs of two continuous functions $f_1, f_2 : (-\infty, a] \rightarrow \mathbb{R}$, where $f_1(a) = f_2(a)$. Let B be another left chain obtained as the union of the graphs of two continuous functions $g_1, g_2 : (-\infty, b] \rightarrow \mathbb{R}$, and assume that A and B are disjoint. We can assume, by symmetry, that $b \leq a$. Consider $g_1(b) = g_2(b)$. It is easy to see that if it is below both $f_1(b)$ and $f_2(b)$, then B is below A . If it is above both $f_1(b)$ and $f_2(b)$, then B is above A . Finally, if $g_1(b)$ is between $f_1(b)$ and $f_2(b)$, then B is to the left of A . A similar argument applies if both A and B are right chains. Finally, if a left chain is disjoint from a right chain, then the left chain is always to the left of the right chain. This completes the proof of Lemma 8.

Now we return to the proof of Theorem 7. Fix a family F of 2-way infinite 3-monotone chains, and let G denote their disjointness graph. Let F_1 and F_2 consist of the wide and non-wide elements of F , respectively. We have seen that the disjointness graph $G[F_1]$ of F_1 is a comparability graph. Comparability graphs are perfect, so we have $\chi(G[F_1]) = \omega(G[F_1])$. We also proved that the comparability graph $G[F_2]$ of F_2 is the union of three comparability graphs. This implies that $\chi(G[F_2]) \leq (\omega(G[F_2]))^3$.

For the entire graph G , we have

$$\chi(G) \leq \chi(G[F_1]) + \chi(G[F_2]) \leq \omega(G[F_1]) + (\omega(G[F_2]))^3 \leq \omega(G) + (\omega(G))^3,$$

as required. This completes the proof of the theorem. ◀

In [25], for every $k \geq 2$, we constructed arrangements of x -monotone curves such that their left endpoints lie on the y -axis and their disjointness graphs have clique number k and chromatic number $\binom{k+1}{2}$. We can extend these curves to the left by adding horizontal half-lines without changing their intersection structure. Traversing the resulting curves twice, we obtain families of 2-way infinite 2-monotone chains such that their disjointness graphs satisfy $\chi(G) = \binom{\omega(G)+1}{2}$.

We were unable to improve on the bound in Theorem 7 even for 2-way infinite polygonal 3-chains. The best lower bound we have in this case is $\omega(G)^{(\log 5 / \log 2)^{-1}} \approx \omega(G)^{1.32}$, and it follows from a construction in [19].

5 Concluding remarks

A. Given an arrangement \mathcal{C} of curves in the plane and a line ℓ , we say that \mathcal{C} is *grounded on* ℓ if every member $c \in \mathcal{C}$ lies in the same closed half-plane bounded by ℓ , and c has precisely one point in common with ℓ , which is one of its endpoints.

The chromatic number of intersection graphs of grounded curves has been extensively studied (see [6], for a survey), but less is known about the corresponding problem for disjointness graphs. In the proof of Theorem 1, we constructed arrangements of 1-way infinite V-shapes whose disjointness graphs are triangle-free and whose chromatic numbers are arbitrarily large. Applying a suitable projective transformation, these arrangements can be turned into arrangements of *grounded* V-shapes.

B. In Problem 2, we asked whether the disjointness graph of an arrangement of V-shapes can have simultaneously arbitrarily high chromatic number and girth. The following statement provides an affirmative answer to a relaxed version of this question. The *odd-girth* of a graph is the length of the shortest odd cycle in it (or infinite if the graph is bipartite).

► **Proposition 9.** *There exist arrangements of polygonal 2-chains in the plane whose disjointness graphs have arbitrarily large odd-girths and chromatic numbers.*

Proof. The proof is based on the same idea as the Proof of Theorem 1, where we represented the shift graph S_n as the disjointness graph of an arrangement of V-shapes. The vertices of S_n are pairs (a, b) of integers $1 \leq a < b \leq n$, so they can be associated with the edges of the complete graph K_n . Thus, the vertices of S_n associated with the edges of a subgraph $G \subseteq K_n$ induce a subgraph $G^* \subseteq S_n$. It is easy to verify that for any $G \subseteq K_n$, we have

- (1) $\chi(G^*) \geq \log(\chi(G))$ and
- (2) the odd-girth of G^* is strictly larger than the odd-girth of G .

For any integers g and k , there exist $n = n(g, k)$ and a subgraph $G \subseteq K_n$ with girth (and, hence, odd-girth) at least g and chromatic number at least k . By properties (1) and (2), the odd-girth of the corresponding induced subgraph G^* of S_n will be larger than g , and its chromatic number will be at least $\log k$. The graph G^* inherits from S_n a representation as a disjointness graph of V-shapes. ◀

Unfortunately, getting rid of short *even* cycles, even 4-cycles, looks impossible by using this simple trick.

C. The arrangements of polygonal curves proving Theorems 1 and 3 have the property that any two of them have at most *two* points in common. It would be interesting to decide whether these theorems remain true if we insist that the curves are *single-crossing*, that is, any two curves have at most one point in common at which they properly cross.

► **Conjecture 10.** *The class of disjointness graphs of single-crossing polygonal 2-chains is χ -bounded.*

Mütze *et al.* [23] proved that the same statement is false for polygonal 3-chains.

D. To prove Theorem 1, we established that the shift graph S_n , a triangle-free graph of unbounded chromatic number, can be obtained as the disjointness graph of V-shapes. However, the *fractional chromatic number* of S_n is bounded: it is smaller than 4 for every n . Do there exist triangle-free disjointness graphs of V-shapes with arbitrarily large fractional chromatic number?

Analogously, our construction for Theorem 3 gives disjointness graphs with bounded fractional chromatic number. Do there exist disjointness graphs of Z-shapes with arbitrarily large girth and fractional chromatic number?

References

- 1 Edgar Asplund and Branko Grünbaum. On a coloring problem. *Mathematica Scandinavica*, 8(1):181–188, 1960.
- 2 Prosenjit Bose, Hazel Everett, Sándor P Fekete, Michael E Houle, Anna Lubiw, Henk Meijer, Kathleen Romanik, Günter Rote, Thomas C Shermer, Sue Whitesides, et al. A visibility representation for graphs in three dimensions. In *Graph Algorithms And Applications I*, pages 103–118. World Scientific, 2002.
- 3 James P. Burling. On coloring problems of families of prototypes. (PhD thesis), University of Colorado, Boulder, 1965.
- 4 Parinya Chalermsook and Bartosz Walczak. Coloring and maximum weight independent set of rectangles. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 860–868. SIAM, 2021.
- 5 James Davies. Box and segment intersection graphs with large girth and chromatic number. *Advances in Combinatorics*, 2021.
- 6 James Davies, Tomasz Krawczyk, Rose McCarty, and Bartosz Walczak. Grounded l-graphs are polynomially chi-bounded. *arXiv preprint*, 2021. [arXiv:2108.05611](https://arxiv.org/abs/2108.05611).
- 7 Michelle Delcourt and Luke Postle. Reducing linear hadwiger’s conjecture to coloring small graphs. *arXiv preprint*, 2021. [arXiv:2108.01633](https://arxiv.org/abs/2108.01633).
- 8 Paul Erdős and András Hajnal. On chromatic number of infinite graphs, theory of graphs. In *Proc. Colloq. Tihany, Hungary*, pages 83–98, 1966.
- 9 Paul Erdős and András Hajnal. On chromatic number of graphs and set-systems. *Acta Math. Acad. Sci. Hungar.*, 17(61-99):1, 1966.
- 10 Peter Frankl. A constructive lower bound for ramsey numbers. *Ars Combinatorica*, 3(297-302):28, 1977.
- 11 Sylvain Gravier, Chinh T Hoang, and Frédéric Maffray. Coloring the hypergraph of maximal cliques of a graph with no long path. *Discrete mathematics*, 272(2-3):285–290, 2003.
- 12 András Gyárfás. On ramsey covering-numbers. *Infinite and Finite Sets*, 2:801–816, 1975.
- 13 András Gyárfás. On the chromatic number of multiple interval graphs and overlap graphs. *Discrete mathematics*, 55(2):161–166, 1985.
- 14 András Gyárfás. Problems from the world surrounding perfect graphs. *Applicationes Mathematicae*, 19(3-4):413–441, 1987.
- 15 András Gyárfás and Jenő Lehel. Hypergraph families with bounded edge cover or transversal number. *Combinatorica*, 3(3-4):351–358, 1983.
- 16 András Gyárfás and Jenő Lehel. Covering and coloring problems for relatives of intervals. *Discrete Mathematics*, 55(2):167–180, 1985.
- 17 Hugo Hadwiger. Über eine klassifikation der streckenkomplexe. *Vierteljschr. Naturforsch. Ges. Zürich*, 88(2):133–142, 1943.

- 18 Navin Kashyap, Paul H Siegel, and Alexander Vardy. An application of ramsey theory to coding for the optical channel. *SIAM Journal on Discrete Mathematics*, 19(4):921–937, 2005.
- 19 David Larman, Jiří Matoušek, János Pach, and Jenő Törőcsik. A ramsey-type result for convex sets. *Bulletin of the London Mathematical Society*, 26(2):132–136, 1994.
- 20 Manor Mendel and Assaf Naor. Ramsey partitions and proximity data structures. *Journal of the European Mathematical Society*, 9(2):253–275, 2007.
- 21 Friedhelm Meyer auf der Heide and Avi Wigderson. The complexity of parallel sorting. *SIAM Journal on Computing*, 16(1):100–107, 1987.
- 22 Burkhard Monien and Ewald Speckenmeyer. Ramsey numbers and an approximation algorithm for the vertex cover problem. *Acta Informatica*, 22(1):115–123, 1985.
- 23 Torsten Mütze, Bartosz Walczak, and Veit Wiechert. Realization of shift graphs as disjointness graphs of 1-intersecting curves in the plane. *arXiv preprint*, 2018. [arXiv:1802.09969](https://arxiv.org/abs/1802.09969).
- 24 János Pach, Gábor Tardos, and Géza Tóth. Disjointness graphs of segments in the space. *Combinatorics, Probability and Computing*, 30(4):498–512, 2021.
- 25 János Pach and István Tomon. On the chromatic number of disjointness graphs of curves. *Journal of Combinatorial Theory, Series B*, 144:167–190, 2020.
- 26 Arkadiusz Pawlik, Jakub Kozik, Tomasz Krawczyk, Michał Lasoń, Piotr Micek, William T Trotter, and Bartosz Walczak. Triangle-free intersection graphs of line segments with large chromatic number. *Journal of Combinatorial Theory, Series B*, 105:6–10, 2014.
- 27 Vera Rosta. Ramsey theory applications. *The Electronic Journal of Combinatorics*, 1000:DS13–Dec, 2004.
- 28 Alex Scott and Paul Seymour. A survey of χ -boundedness. *Journal of Graph Theory*, 95(3):473–504, 2020.
- 29 David P Sumner. Subtrees of a graph and chromatic number. *The Theory and Applications of Graphs*, (G. Chartrand, ed.), John Wiley & Sons, New York, 557:576, 1981.

Covering Points by Hyperplanes and Related Problems

Zuzana Patáková  

Department of Algebra, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Micha Sharir  

School of Computer Science, Tel Aviv University, Tel Aviv, Israel

Abstract

For a set P of n points in \mathbb{R}^d , for any $d \geq 2$, a hyperplane h is called k -rich with respect to P if it contains at least k points of P . Answering and generalizing a question asked by Peyman Afshani, we show that if the number of k -rich hyperplanes in \mathbb{R}^d , $d \geq 3$, is at least $\Omega(n^d/k^\alpha + n/k)$, with a sufficiently large constant of proportionality and with $d \leq \alpha < 2d - 1$, then there exists a $(d - 2)$ -flat that contains $\Omega(k^{(2d-1-\alpha)/(d-1)})$ points of P . We also present upper bound constructions that give instances in which the above lower bound is tight. An extension of our analysis yields similar lower bounds for k -rich spheres.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Rich hyperplanes, Incidences, Covering points by hyperplanes

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.57

Funding Zuzana Patáková: Work partially supported by Charles University projects UNCE/SCI/022 and PRIMUS/21/SCI/014.

Micha Sharir: Work partially supported by ISF Grant 260/18.

Acknowledgements The authors thank Peyman Afshani for sharing his thoughts with us concerning this problem.

1 Introduction

Let P be a set of n points in \mathbb{R}^d . A hyperplane h is called k -rich with respect to P if it contains at least k points of P . Assume that the number of k -rich hyperplanes is at least $\Omega(n^d/k^{d+1} + n/k)$, with a sufficiently large constant of proportionality. Is there a lower-dimensional flat containing “a lot of points” of P ? This question was raised by Peyman Afshani (personal communication), motivated by his recent work [1] on point covering problems. We answer Afshani’s problem in the affirmative, in the following stronger form.

► **Theorem 1.** *Let $d \geq 3, k \geq d$ be integers, and $d \leq \alpha < 2d - 1$. Let P be a set of n points in \mathbb{R}^d , for which the number of k -rich hyperplanes is at least $c(n^d/k^\alpha + n/k)$, for some sufficiently large constant c (depending only on d). Then there exists a $(d - 2)$ -flat that contains $\Omega(k^{(2d-1-\alpha)/(d-1)})$ points of P .*

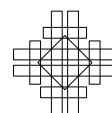
We also present two upper bound constructions that give instances of the problem in which the bound in Theorem 1 is tight. The first instance involves $\alpha = d + 1$ (as in Afshani’s original question) and certain values of k , and in the second instance we have $\alpha = d = 3$.

We also extend our analysis to the case of k -rich spheres (spheres that contain at least k points of P). We show (see Theorem 4) that if the number of k -rich $(d - 1)$ -spheres is at least $c(n^{d+1}/k^\alpha + n/k)$, for $d + 1 \leq \alpha < 2d + 1$ and for some sufficiently large constant c , then there exists a $(d - 2)$ -sphere that contains $\Omega(k^{(2d+1-\alpha)/d})$ points of P .



© Zuzana Patáková and Micha Sharir;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 57; pp. 57:1–57:7
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



The result is interesting by itself, but it may also find a potential application in the so called *hyperplane cover* problem, one of the classical problems in computational complexity: given a set S of n points in \mathbb{R}^d and a number h , can we find h hyperplanes that cover all points of S ? It is a geometric variant of a set cover problem, and it was shown that already for $d = 2$ the hyperplane cover problem is both NP-hard [13] and APX-hard [11]. However, several FPT-algorithms (in the fixed parameter k) are known, best of which is [16]. In the special cases $d = 2$ and $d = 3$ it has been further improved [1]. The improvement is based on incidence bounds and builds on a simple observation that given a hyperplane cover of cardinality h , some of the hyperplanes might contain many more points than the others. The main idea is to deal with such hyperplanes first and the performance of the algorithm depends on the number of such hyperplanes. For example, it follows from the Szemerédi-Trotter theorem that there are at most $O(n^d/k^3)$ k -rich hyperplanes defined by n points in \mathbb{R}^d . However, in the approach of [1] this bound turned out to be useful only in the plane, in which case the exponent of n is strictly smaller than the exponent of k . The 3-dimensional case is treated using another incidence bound [9], but this approach also does not extend to higher dimensions [1]. What could help for $d \geq 4$ is to show that if there are too many rich hyperplanes the points cannot be distributed arbitrarily, in fact, many of them must lie on a common lower dimensional flat. The results of our paper address this issue.

The problem is also closely related to the problem of bounding the number of incidences between n points and m hyperplanes, and we will indeed use tools from incidence theory to tackle this problem. A major hurdle in obtaining sharp point-hyperplane incidence bounds, in $d \geq 3$ dimensions, is the possibility that there exists a $(d - 2)$ -flat that contains many of the points and is contained in many of the hyperplanes. In the worst case all the n points could be contained in such a flat, and all the m hyperplanes could contain the flat, and then the number of incidences would be nm , the largest possible value. To obtain sharper bounds one usually needs to require that no $(d - 2)$ -flat contains too many points, or that is not contained in too many hyperplanes, or to impose other restrictions on the setup. See [2, 4, 5, 7, 9, 14, 15] for a sample of earlier works on this topic. For example, better bounds can be obtained if the points are restricted to be vertices of the arrangement of the hyperplanes [2], or when the incidence graph between the points and hyperplanes does not contain a complete bipartite subgraph of some small size (see [5]). Improved bounds can also be obtained by assuming that no lower-dimensional flat is contained in too many hyperplanes, or does not contain too many points [7]. Some of these works also derive lower bounds, but for different quantities, which do not seem directly related to the setup considered in this paper. See, for example, Apfelbaum and Sharir [4] and Brass and Knauer [5] for lower bounds on the maximum size of a complete bipartite subgraph in the incidence graph of points and hyperplanes.

2 Proof of Theorem 1

Let P be a set of n points in \mathbb{R}^d that has many k -rich hyperplanes, in the sense of Theorem 1, and let ℓ denote the maximum number of points of P contained in any $(d - 2)$ -flat. We seek a lower bound on ℓ .

Overview of the proof. Before we dive into the details, we describe the overall idea first. Let H be the set of all k -rich hyperplanes spanned by P . By a simple argument we show that H is finite and then we establish a lower and an upper bound on the number of incidences between P and H . Comparing these bounds yields the desired result. As the lower bound

on the number of incidences is trivially $k|H|$, the actual work here is to obtain a reasonable upper bound – for that we use simplicial partitions (Theorem 3), point-hyperplane duality, and the Cauchy-Schwartz inequality.

We start with a simple incidence bound.

► **Lemma 2.** *Let P and H be finite sets of points and hyperplanes in \mathbb{R}^d , respectively. We have the following simple bound on the number $I(P, H)$ of incidences between the points of P and the hyperplanes of H .*

$$I(P, H) = O(|H||P|^{1/2}\ell^{1/2} + |P|). \quad (1)$$

Proof. This is a simple geometric application of the well known Kővári-Sós-Turán Theorem (see, e.g., [3, 10]), which says that a $K_{\ell,2}$ -free bipartite graph with n left and m right vertices has at most $O(mn^{1/2}\ell^{1/2} + n)$ edges. The proof is based on the observation that the incidence graph between P and H does not contain $K_{\ell+1,2}$ as a subgraph. Indeed, any pair of non-parallel hyperplanes from H intersect in a $(d-2)$ -flat, which, by assumption, contains at most ℓ points of P . ◀

Using simplicial partitions. We now proceed to sharpen the upper bound in Lemma 2. We recall the following result, due to Matoušek [12].

► **Theorem 3.** *Let Q be a set of m points in \mathbb{R}^d , for any $d \geq 2$, and let $1 < r \leq m$ be a given parameter. Then Q can be partitioned into $q \leq 2r$ subsets, Q_1, \dots, Q_q , so that, for each i , $m/(2r) \leq |Q_i| \leq m/r$, and Q_i is contained in the relative interior of a (possibly lower-dimensional) simplex Δ_i , so that every hyperplane crosses (i.e., intersects but does not contain) at most $O(r^{1-1/d})$ of these simplices.*

The partition in Theorem 3 is referred to as a *simplicial partition* of Q . We remark that the theorem guarantees that none of the simplices is a single point when $r \leq m/4$. This result has more recently been refined by Chan [6], but the original version suffices for our purpose.

Proof of Theorem 1. First note that if there is a $(d-2)$ -flat containing at least k points of P , the theorem trivially holds, as we then have $\ell \geq k \geq k^{(2d-1-\alpha)/(d-1)}$, since $\alpha \geq d$. Hence we can assume that each $(d-2)$ -flat contains at most $k-1$ points of P . This guarantees that the number of all k -rich hyperplanes (with respect to P) is finite, as every k -tuple of points of P spans at most one k -rich hyperplane.

Let then H be the finite set of all k -rich hyperplanes, $k \geq d$, set $m := |H|$, and recall that we assume that $m = |H| \geq c(n^d/k^\alpha + n/k)$, for some sufficiently large constant c (that depends on d) and for some $d \leq \alpha < 2d-1$.

Our strategy is to derive an upper bound on the number of incidences between the points of P and the hyperplanes of H , and combine it with the obvious lower bound mk on this number, which follows since each of these hyperplanes is k -rich. A combination of these bounds will lead to the desired lower bound on ℓ .

We apply standard geometric duality in \mathbb{R}^d and get a set H^* of m dual points and a set P^* of n dual hyperplanes. The dual version of the fact that no $(d-2)$ -flat contains more than ℓ points of P is that no line is contained in more than ℓ hyperplanes of P^* . We also know, as just mentioned, that $I(P, H) \geq mk$, as each primal hyperplane in H contains at least k points of P .

We fix some r , which we determine later, and apply Theorem 3 in the dual setting. We obtain $q \leq 2r$ subsets H_1^*, \dots, H_q^* , so that $m/(2r) \leq |H_i^*| \leq m/r$ for each $i = 1, \dots, q$, and each hyperplane crosses $O(r^{1-1/d})$ of the corresponding simplices. Denote also by P_i^* the set of dual hyperplanes that cross the i -th simplex $\Delta_i \supset H_i^*$, for each i . Let P_i and H_i denote the corresponding sets of points and hyperplanes in the primal space.

57:4 Covering Points by Hyperplanes and Related Problems

The number of incidences of dual points inside the partition cells and dual hyperplanes crossing the corresponding simplices can be bounded as follows:

$$\begin{aligned} \sum_{i=1}^q I(H_i^*, P_i^*) &= \sum_{i=1}^q I(P_i, H_i) = O\left(\sum_{i=1}^q |P_i|^{1/2} |H_i| \ell^{1/2} + \sum_{i=1}^q |P_i|\right) \\ &= O\left(m(\ell n)^{1/2} r^{-1/(2d)} + nr^{1-1/d}\right). \end{aligned} \quad (2)$$

The first inequality follows by applying the bound (1) of Lemma 2 in the primal. For the second inequality we use the property that each dual hyperplane crosses at most $O(r^{1-1/d})$ cells, so we have, using the Cauchy-Schwarz inequality, and recalling that $q \leq 2r$,

$$\begin{aligned} |H_i| = |H_i^*| &\leq \frac{m}{r}, \quad \sum |P_i| = \sum |P_i^*| = O(r^{1-1/d}n), \quad \text{and} \\ \sum_{i=1}^q |P_i|^{1/2} &\leq \left(\sum_{i=1}^q |P_i|\right)^{1/2} (2r)^{1/2} = O(n^{1/2} r^{(2d-1)/(2d)}), \end{aligned}$$

and the second inequality follows.

It remains to count the incidences between points in a cell (simplex) and hyperplanes that contain the simplex. Any such simplex σ is j -dimensional, for some $1 \leq j \leq d-1$ (zero-dimensional simplices do not arise when $r \leq m/4$). When $j = d-1$, each such σ is contained in at most one hyperplane of P^* , contributing in total at most m' incidences, where m' is the number of dual points contained in such cells. When $1 \leq j \leq d-2$, σ spans (affinely) a j -flat g , which cannot be contained in more than ℓ dual hyperplanes in P^* , for otherwise any line in g would also be contained in these hyperplanes, contrary to our assumption. Hence the number of resulting incidences is at most $\ell m''$, where m'' is the number of dual points contained in such simplices. In total, all the lower-dimensional simplices contribute at most ℓm incidences.

Hence, combining this with (2), we get:

$$mk \leq I(P, H) \leq O\left(m\ell^{1/2}n^{1/2}r^{-1/(2d)} + r^{1-1/d}n\right) + \ell m. \quad (3)$$

We now balance the first two terms by choosing

$$r := \left(\frac{\ell m^2}{n}\right)^{d/(2d-1)}.$$

For this to make sense r has to be between 1 and $m/4$. We note that $r < 1$ when $m < (n/\ell)^{1/2}$ and $r > m/4$ when $m > c_1 n^d/\ell^d$, for some constant c_1 that depends on d . In the former case we take $r = 1$ and the first two terms become $O(n)$. (Note that the choice $r = 1$ corresponds to a direct application of Lemma 2.) In the latter case we take $r = m/4$ and the first two terms become

$$O(m^{(2d-1)/(2d)} \ell^{1/2} n^{1/2} + m^{1-1/d}n) = O\left(m^{(2d-1)/(2d)} \ell^{1/2} n^{1/2}\right) = O(m\ell),$$

where both inequalities hold because $m > c_1 n^d/\ell^d$. When neither of these two extreme cases occurs, the first two terms become $O(m^{(2d-2)/(2d-1)} \ell^{(d-1)/(2d-1)} n^{d/(2d-1)})$. Altogether we thus get

$$mk \leq O\left(m^{(2d-2)/(2d-1)} \ell^{(d-1)/(2d-1)} n^{d/(2d-1)} + m\ell + n\right). \quad (4)$$

The inequality in (4) implies that either $\ell = \Omega(k) = \Omega(k^{(2d-1-\alpha)/(d-1)})$, since $\alpha \geq d$, or

$$m = O\left(\frac{\ell^{d-1}n^d}{k^{2d-1}} + \frac{n}{k}\right), \tag{5}$$

where we have distinguished two cases depending on whether the first or the last term in the right-hand side of (4) dominates. Let c' be the O-notation constant from (5). Since we assume that $m \geq c(n^d/k^\alpha + n/k)$, where c is a sufficiently large constant, we get

$$c\left(\frac{n^d}{k^\alpha} + \frac{n}{k}\right) \leq c'\left(\frac{\ell^{d-1}n^d}{k^{2d-1}} + \frac{n}{k}\right).$$

For $c \geq c'$ it simplifies to

$$\frac{cn^d}{k^\alpha} \leq \frac{cn^d}{k^\alpha} + (c - c')\frac{n}{k} \leq c'\frac{\ell^{d-1}n^d}{k^{2d-1}},$$

which implies that $\ell = \Omega(k^{(2d-1-\alpha)/(d-1)})$. This completes the proof of Theorem 1. ◀

2.1 Upper bound constructions

First construction. The following construction only handles the case $\alpha = d + 1$ (the original question of Afshani) and certain restricted values of k ; it is a variation of a construction of Elekes [8].

Fix two integer parameters $u > v \geq 1$ where v is a suitable constant. Let P be the set of vertices of the $u \times \dots \times u \times duv$ integer grid in \mathbb{R}^d . That is,

$$P = \{(i_1, \dots, i_d) \mid 0 \leq i_1, \dots, i_{d-1} \leq u - 1, 0 \leq i_d \leq duv - 1\}.$$

We have $n := |P| = du^dv$ and we set $k := u^{d-1}$. Any hyperplane of the form $x_d = a_1x_1 + a_2x_2 + \dots + a_{d-1}x_{d-1} + a_d$, with integer coefficients satisfying $0 \leq a_i \leq v - 1$, for $1 \leq i \leq d - 1$, and $0 \leq a_d \leq uv - 1$, is trivially seen to be k -rich with respect to P . Hence the number of k -rich hyperplanes is at least uv^d . On the other hand, we have

$$\frac{n^d}{k^{d+1}} = \frac{d^d u^{d^2} v^d}{u^{(d+1)(d-1)}} = d^d uv^d.$$

It is easily verified that a $(d - 2)$ -flat λ that is not vertical (i.e., not parallel to the x_d -axis) contains at most u^{d-2} points of P , and that a vertical $(d - 2)$ -flat can contain $u^{d-3}duv = O(u^{d-2}) = O(k^{(d-2)/(d-1)})$ points of P (but not more). Hence, setting ℓ to be $ck^{(d-2)/(d-1)}$, for a suitable coefficient c , we have a construction with at least $\frac{n}{d^d k^{d+1}}$ k -rich hyperplanes, but no $(d - 2)$ -flat contains more than $ck^{(d-2)/(d-1)}$ points of P . In other words, our bound is asymptotically worst-case tight for this special setup.

We remark that in this construction we have $k = \Theta(n^{1-1/d})$, so one still needs to show that the bound is tight for other values of k . We leave this as an open problem.

Second construction. A more significant open challenge is to extend the construction to other values of α in the range $d \leq \alpha < 2d - 1$. We make a first step towards this goal, by presenting, for $\alpha = d = 3$, another simple construction. Let $k \geq 3, k \geq u \geq 2$ be integer parameters. Consider a set L of u pairwise skew lines in \mathbb{R}^3 , each containing k distinguished points. Let P be the set of all these points. We have $n := |P| = ku$. Note that there are infinitely many k -rich planes with respect to P as any plane containing a single line from L

is k -rich. On the other hand, it follows from the construction that no line contains strictly more than k points of P . Indeed, any line not contained in L intersects at most k lines from L (since $u \leq k$), so it can contain at most k points of P . Hence, $\ell = k$, which shows that the bound in Theorem 1 is tight for $d = \alpha = 3$ and $n/2 \geq k \geq n^{1/2}$.

3 The case of spheres

The analysis can be extended to the case of spheres in a straightforward manner. Specifically, we have a set P of n points in \mathbb{R}^d , for $d \geq 3$. We say that a sphere σ is k -rich with respect to P if it contains at least k points of P . The goal now is to show that if there are many k -rich $(d-1)$ -spheres in \mathbb{R}^d then there exists a $(d-2)$ -sphere that contains many points of P . The concrete statement is:

► **Theorem 4.** *Let $d \geq 3, k \geq d+1$ be integers, and let $d+1 \leq \alpha < 2d+1$ be a parameter. Let P be a set of n points in \mathbb{R}^d , for which the number of k -rich $(d-1)$ -spheres is at least $c(n^{d+1}/k^\alpha + n/k)$, for some sufficiently large constant c . Then there exists a $(d-2)$ -sphere that contains $\Omega(k^{(2d+1-\alpha)/d})$ points of P .*

Note that if there is a k -rich $(d-2)$ -sphere, the theorem holds trivially, as we then have $\ell \geq k$ and $\alpha \geq d+1$. Hence we can assume that no $(d-2)$ -sphere is k -rich, which implies, as in the case of hyperplanes, that the number of k -rich $(d-1)$ -spheres is finite.

The proof is an adaptation of the preceding analysis. Let P be a set of n points in \mathbb{R}^d , for $d \geq 3$, that has many k -rich $(d-1)$ -spheres, in the sense of Theorem 4. Let ℓ denote the maximum number of points of P contained in any $(d-2)$ -sphere. As before, we seek a lower bound on ℓ .

Lemma 2 continues to hold in the case of spheres, with more or less the same proof, using the obvious property that two non-disjoint $(d-1)$ -spheres intersect in a $(d-2)$ -sphere or a single point. To sharpen the bound we proceed as follows.

Let Σ be the set of all k -rich $(d-1)$ -spheres, $k \geq d+1$, and recall that we assume that $m := |\Sigma| \geq c(n^{d+1}/k^\alpha + n/k)$, for some sufficiently large constant c (that depends on d) and for $d+1 \leq \alpha < 2d+1$.

We apply the standard lifting transform $(x_1, \dots, x_d) \mapsto (x_1, \dots, x_d, x_1^2 + \dots + x_d^2)$, which transforms $(d-1)$ -spheres in \mathbb{R}^d to hyperplanes in \mathbb{R}^{d+1} . Applying standard duality in \mathbb{R}^{d+1} , we get a set Σ^* of m dual points and a set P^* of n dual hyperplanes in \mathbb{R}^{d+1} . The lifted-dual version of the fact that no $(d-2)$ -sphere, which is lifted to a $(d-1)$ -flat in \mathbb{R}^{d+1} , contains more than ℓ points of P is that no line is contained in more than ℓ hyperplanes of P^* . As in the case of rich hyperplanes, we also know that $I(P, \Sigma) \geq mk$.

In other words, after this transform we reach the same problem involving points and hyperplanes in \mathbb{R}^{d+1} , and we can apply the preceding analysis verbatim with $d+1$ replacing d , and obtain the assertion in Theorem 4.

4 Discussion

The problem studied in this work can be considered as a variant in the study of incidences between points and hyperplanes. As far as we can tell, the results in the previous works that have studied such problems (e.g., [4, 5]) do not imply our results.

Several open problems arise. For example, are there variants of our assumptions, in $d \geq 4$ dimensions, that imply the existence of an even lower-dimensional flat that contain many points of P ? This does not hold without any further assumptions, because we can place the

points of P in *general position* in some $(d - 2)$ -flat g , and then there are infinitely many k -rich hyperplanes, for any k (all hyperplanes that contain g), but no $(d - 3)$ -flat contains more than $d - 2$ points of P .

Other problems, already mentioned earlier, are to obtain upper bound constructions, other than the one in Section 2.1, for other values of k and of α .

References

- 1 P. Afshani, E. Berglin, I. van Duijn, and J. S. Nielsen. Applications of incidence bounds in point covering problems. In *Proc. 32nd ACM Sympos. Comput. Geom.*, pages 60:1–60:15, 2016.
- 2 P. K. Agarwal and B. Aronov. Counting facets and incidences. *Discrete Comput. Geom.*, 7:359–369, 1992.
- 3 P. K. Agarwal and J. Pach. *Combinatorial Geometry*. Wiley-Interscience, NY, 1995.
- 4 R. Apfelbaum and M. Sharir. Large bipartite graphs in incidence graphs of points and hyperplanes. *SIAM J. Discrete Math.*, 21:707–725, 2007.
- 5 P. Brass and C. Knauer. On counting point-hyperplane incidences. *Comput. Geom. Theory Appl.*, 25:13–20, 2003.
- 6 T. M. Chan. Optimal partition trees. *Discrete Comput. Geom.*, 47:661–690, 2012.
- 7 H. Edelsbrunner, L. Guibas, and M. Sharir. The complexity and construction of many faces in arrangements of lines and of segments. *Discrete Comput. Geom.*, 5:61–196, 1990.
- 8 G. Elekes. Sums versus products in number theory, algebra and Erdős geometry – a survey. In *Paul Erdős and his Mathematics II*, volume 11, pages 241–290. Bolyai Math. Soc. Stud., 2002.
- 9 G. Elekes and C. Tóth. Incidences of not-too-degenerate hyperplanes. In *Proc. 21st ACM Sympos. Comput. Geom.*, pages 13–20, 2005.
- 10 T. Kővari, V. T. Sós, and P. Turán. On a problem of K. Zarankiewicz. *Colloq. Math.*, 3:50–57, 1954.
- 11 V. S. A. Kumar, S. Arya, and H. Ramesh. Hardness of set cover with intersection 1. In *Automata, languages and programming (Geneva, 2000)*, volume 1853 of *Lecture Notes in Comput. Sci.*, pages 624–635. Springer, Berlin, 2000.
- 12 J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.*, 8:315–334, 1992.
- 13 N. Megiddo and A. Tamir. On the complexity of locating linear facilities in the plane. *Oper. Res. Lett.*, 1(5):194–197, 1982.
- 14 M. Rudnev. On the number of incidences between points and planes in three dimensions. *Combinatorica*, 38:219–254, 2018.
- 15 N. Singer and M. Sudhan. Point-hyperplane incidence geometry and the log-rank conjecture. [arXiv:2101.09592](https://arxiv.org/abs/2101.09592).
- 16 J. Wang, W. Li, and J. Chen. A parameterized algorithm for the hyperplane-cover problem. *Theor. Comput. Sci.*, 411:4005–4009, 2010.

The Degree-Rips Complexes of an Annulus with Outliers

Alexander Rolle  

Department of Mathematics, Technische Universität München, Germany

Abstract

The degree-Rips bifiltration is the most computable of the parameter-free, density-sensitive bifiltrations in topological data analysis. It is known that this construction is stable to small perturbations of the input data, but its robustness to outliers is not well understood. In recent work, Blumberg–Lesnick prove a result in this direction using the Prokhorov distance and homotopy interleavings. Based on experimental evaluation, they argue that a more refined approach is desirable, and suggest the framework of homology inference. Motivated by these experiments, we consider a probability measure that is uniform with high density on an annulus, and uniform with low density on the disc inside the annulus. We compute the degree-Rips complexes of this probability space up to homotopy type, using the Adamaszek–Adams computation of the Vietoris–Rips complexes of the circle. These degree-Rips complexes are the limit objects for the Blumberg–Lesnick experiments. We argue that the homology inference approach has strong explanatory power in this case, and suggest studying the limit objects directly as a strategy for further work.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases multi-parameter persistent homology, stability, homology inference

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.58

Supplementary Material *Software (Source Code)*: https://github.com/alexanderrolle/degree_Rips_annulus; archived at `swh:1:dir:fa545846978022bdf95b9e2f53f5be18685d2882`

Acknowledgements I would like to thank Michael Lesnick for helpful conversations about robustness of degree-Rips, and Luis Scoccola and Fabian Roll for various helpful conversations about topics related to this paper. I would also like to thank the reviewers for their constructive comments.

1 Introduction

1.1 Background

The degree-Rips bifiltration [15] is a density-sensitive construction based on the Vietoris–Rips filtration. The sensitivity to density has two consequences: degree-Rips can distinguish metric spaces that are close in the Gromov–Hausdorff distance but have different patterns of density, and degree-Rips is more robust to noise and outliers. There are other bifiltrations that share these goals, but degree-Rips is of particular interest because, using available algorithms and software, it is the most computable of these bifiltrations that requires only a metric on the data as input.

If X is a finite metric space, the degree-Rips complex $\text{DR}(X)$, at parameter (s, k) , is the full subcomplex of the Vietoris–Rips complex $\text{VR}(X)(s)$ on those vertices having degree at least $k - 1$ in the one-skeleton. Equivalently, we take the Vietoris–Rips complex of the subset $X_{(s,k)} = \{x \in X : |B(x, s)| \geq k\}$, where $B(x, s)$ is the open ball in X about x of radius s .

There has now been work on the stability of degree-Rips by several authors. Recent results of Blumberg–Lesnick [4] are notable in that they allows for true outliers: one can add an arbitrary point to a finite metric space, and their results guarantee some relationship between the respective degree-Rips bifiltrations. The main result of Blumberg–Lesnick for degree-Rips says that if the Gromov–Prokhorov distance between the uniform probability



© Alexander Rolle;

licensed under Creative Commons License CC-BY 4.0

38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 58; pp. 58:1–58:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



measures of two finite metric spaces is less than δ , then one has a homotopy interleaving between their degree-Rips bifiltrations, with additive term δ , and with a multiplicative factor in the Rips parameter s . They show moreover that the multiplicative factor is tight.

This framework for studying the robustness of degree-Rips is very natural, but in the same paper, Blumberg–Lesnick observe that the result does not fully capture the robustness of degree-Rips observed in practice. They report on the following experiment. They consider two pointclouds: a uniform sample of 475 points from an annulus, and another pointcloud obtained by adding 25 points sampled uniformly from the disc inside the annulus. Then they use the RIVET software [22] to visualize $H_1\text{DR}$ of both pointclouds. Given the output on the sample with no outliers, their results guarantee that a certain region of the degree-Rips parameter space for the sample with outliers must have non-zero Hilbert function (i.e., the degree-Rips complexes in this region must have non-zero H_1); however this region is small compared to the observed region where the Hilbert function is non-zero. It appears that there is a trade-off between the generality of this result, and the ability to provide explanatory power in concrete cases such as this one.

We make one more remark before explaining the contribution of this paper. The degree-Rips bifiltration is closely related to existing methods for clustering. Several widely-used algorithms that arose independently of topological data analysis, such as the hierarchical clustering algorithm robust single-linkage [8] and the clustering algorithms DBSCAN [11] and HDBSCAN [7], can be computed directly from degree-Rips by taking the connected components of the 1-skeleton. These algorithms are used in part because of their observed robustness to noise and outliers. A satisfactory understanding of the robustness of degree-Rips would also add to our understanding of the robustness of these algorithms.

1.2 Homology inference for degree-Rips

Motivated by their experiments, Blumberg–Lesnick suggest the framework of homology inference for obtaining more refined results about the robustness of degree-Rips. We now explain one approach to homology inference for degree-Rips.

There is a natural generalization of the degree-Rips complexes to metric probability spaces (Definition 3). Given such a space (X, μ) , the degree-Rips complex $\text{DR}(X, \mu)$ at parameter (s, k) is the Vietoris–Rips complex of the subset $X_{(s,k)} = \{x \in X : \mu(B(x, s)) \geq k\}$ (Definition 5). If one gives a finite metric space its uniform probability measure, then this definition agrees with the previous one up to normalization. Furthermore, on compact metric probability spaces, degree-Rips is 2-Lipschitz, comparing the input using the Gromov–Hausdorff–Prokhorov distance, and comparing the output using the homotopy-interleaving distance [21, Theorem 6.5.1]. For the sake of this paper, it is not necessary to know the definition of the Gromov–Hausdorff–Prokhorov distance, but just the following consequence. Say that μ is a compactly-supported probability measure on Euclidean space with support C , let X be a finite sample from μ , let μ_X be the uniform measure on X , and let $\bar{\mu}_X$ be the empirical measure on Euclidean space determined by X . If the Hausdorff distance $d_H(X, C)$ and the Prokhorov distance $d_P(\bar{\mu}_X, \mu)$ are less than ϵ , then the homotopy interleaving distance between $\text{DR}(X, \mu_X)$ and $\text{DR}(C, \mu)$ is less than 2ϵ . Here, the hypothesis is stronger than in the result of Blumberg–Lesnick, because it includes the Hausdorff hypothesis, and the conclusion is also stronger, since one obtains additive interleavings. So, we know the limit objects for degree-Rips: in probability, $\text{DR}(X, \mu_X)$ converges to $\text{DR}(C, \mu)$ in the homotopy-interleaving distance as the size of X goes to infinity.

What consequence does this have for the robustness of degree-Rips? One way to pose the question of the robustness of degree-Rips is the following. If we have a finite metric space X , and X' has been obtained from X by adding a small number of outliers, how do we expect

that $\text{DR}(X)$ and $\text{DR}(X')$ are related? Roughly speaking, this is how Blumberg–Lesnick ask the question. On the level of metric probability spaces, there is an analogous question: if we have μ and C as before, and μ' has been obtained from μ by mixing with the uniform measure on some C' with $C \subset C'$, how are $\text{DR}(C, \mu)$ and $\text{DR}(C', \mu')$ related?

Consider the metric probability spaces from which the finite input in the Blumberg–Lesnick experiments are sampled. Let $\mathcal{A}(R, Q, w)$ be the metric probability space that consists of the union of the annulus $\{p \in \mathbb{R}^2 : R \leq \|p\| \leq Q\}$ and the disc $\{p \in \mathbb{R}^2 : \|p\| < R\}$, with a uniform measure on each piece, such that the measure of the disc is equal to w . If $w = 0$, take the underlying metric space to be just the annulus. See Section 2 for a detailed definition. In this paper, we compute the degree-Rips bifiltrations of $\mathcal{A}(R, Q, w)$ up to homotopy type, using the Adamaszek–Adams computation of the Vietoris–Rips complexes of the circle [1]. We now state the result in the case $w > 0$. See Figure 1 for an illustration.¹

► **Theorem 1.** *Let $\mathcal{A}(R, Q, w)$ be a weighted annulus with $w > 0$. There are continuous maps $\varphi_\ell: (0, \infty) \rightarrow [0, 1]$ for $\ell = 0, 1, 2, \dots, \infty$ such that, for any $s > 0$ and any $k \in [0, 1]$,*

$$\text{DR}(\mathcal{A}(R, Q, w))(s, k) \simeq \begin{cases} \emptyset & \text{if } k > \varphi_0(s) \\ S^{2\ell+1} & \text{if } \varphi_\ell(s) > k > \varphi_{\ell+1}(s) \text{ for } \ell \neq \infty \\ * & \text{if } \varphi_\infty(s) > k \end{cases}$$

Moreover, if $0 < \ell < \infty$ and $0 < s \leq s'$ and $0 \leq k' \leq k \leq 1$ are such that

$$\varphi_\ell(s) > k > \varphi_{\ell+1}(s) \quad \text{and} \quad \varphi_\ell(s') > k' > \varphi_{\ell+1}(s'),$$

then the inclusion $\text{DR}(\mathcal{A}(R, Q, w))(s, k) \hookrightarrow \text{DR}(\mathcal{A}(R, Q, w))(s', k')$ is a homotopy equivalence.

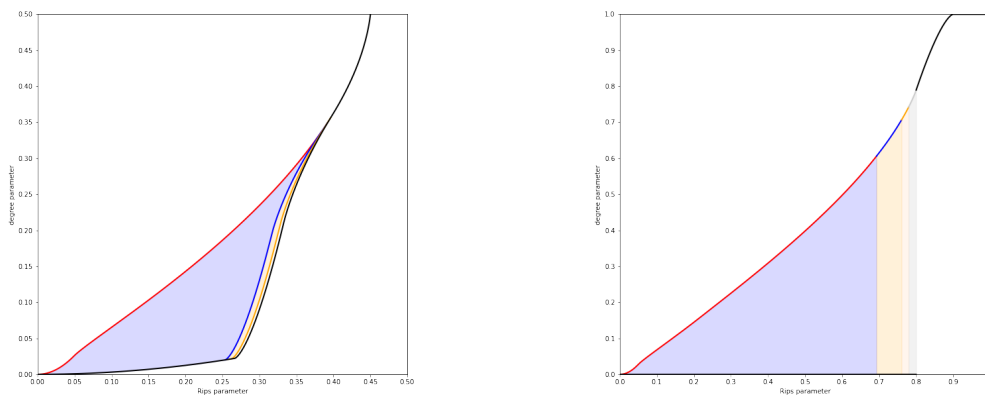
The result for the case $w = 0$ is similar, but the curves that bound the regions are no longer continuous; we state the result in this case in Section 5. Varying w does not have much effect on $\text{DR}(\mathcal{A}(R, Q, w))$ while w remains small and non-zero. Setting $w = 0$ has a large effect, as we see in Figure 1, because in this case only points on the annulus are allowed to appear as vertices of degree-Rips.

Comparing these calculations with the results obtained from finite samples, we see that the homology inference approach indeed provides strong explanatory power. See Figure 2. The region of the parameter space where $\text{DR}(\mathcal{A}(R, Q, w))$ has the homotopy type of S^1 , and thus has rank 1 homology in dimension 1, is similar to the region where the Hilbert function of the degree-Rips complexes of the sample is equal to 1. Note that when we set $w > 0$ and allow for outliers, the region where we see non-zero Hilbert function in the sample extends a little further in the direction of increasing Rips parameter value. For larger values of the Rips parameter, outliers begin to appear as vertices in degree-Rips, and they create connections between dense regions that would not otherwise appear. In $\mathcal{A}(R, Q, w)$, all points in the inner disc are allowed to appear as vertices, and so these connections appear as soon as possible.

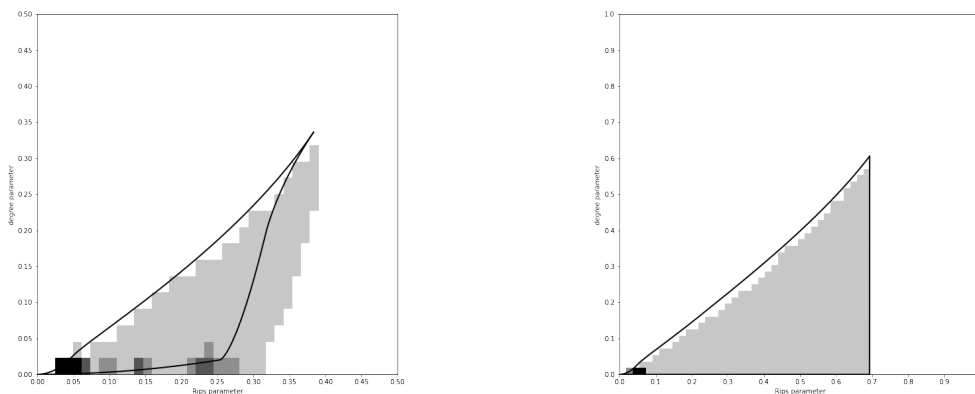
1.3 Related work

Along with the results mentioned in the introduction, Jardine has proved a stability result for degree-Rips [14], using a hypothesis involving configuration spaces, rather than a distance between pointclouds. Much work has been done on homology inference, using a variety of approaches. See for example [3, 17, 2, 18, 5, 6]. The connection between degree-Rips and existing clustering methods was observed by McInnes–Healy [16], and studied further in [13, 20]. There is a large literature on consistency of density-based clustering methods. See for example [10, 8, 19, 9].

¹ Scripts to reproduce the figures are available at https://github.com/alexanderrolle/degreeRips_annulus



■ **Figure 1** On the left, we consider $\mathcal{A}(R, Q, w)$ with inner radius $R = 0.4$, outer radius $Q = 0.5$, and $w = 0.05$, so one expects 25 outliers in a sample of 500, as in the Blumberg–Lesnick experiments. We plot φ_0 (red), φ_1 (blue), φ_2 (yellow), and φ_∞ (black). The blue region is where the homotopy type of $\text{DR}(\mathcal{A}(R, Q, w))$ is S^1 , and the yellow region is where the homotopy type is S^3 . On the right, we consider $\mathcal{A}(R, Q, 0)$. The meaning of the colors is the same. In this case the boundary curves are not continuous. Note that the two figures are plotted at different scales.



■ **Figure 2** We reproduce the Blumberg–Lesnick experiment. On the left we consider a sample with outliers: we sample 500 points from $\mathcal{A}(R, Q, w)$, where R , Q , and w are as in Figure 1. We use RIVET to compute the Hilbert function of $H_1\text{DR}$ of this sample; the light grey region is where the Hilbert function is equal to 1, and darker grey corresponds to higher values of the Hilbert function. The border of the S^1 region of $\text{DR}(\mathcal{A}(R, Q, w))$ is overlaid in black. On the right we consider a sample without outliers: we sample 500 points from $\mathcal{A}(R, Q, 0)$. We compare the Hilbert function of $H_1\text{DR}$ of the sample with the S^1 region of $\text{DR}(\mathcal{A}(R, Q, 0))$ in the same way. Note that the two figures are plotted at different scales.

2 Preliminaries

We now give definitions and conventions that are used throughout the paper. For real numbers a and b , the statement $a < b$ implies $a \neq b$.

► **Definition 2.** Let X be a metric space and $s > 0$. The **Vietoris–Rips complex** $\text{VR}(X)(s)$ is the simplicial complex

$$\text{VR}(X)(s) = \{\{x_0, \dots, x_n\} \mid d_X(x_i, x_j) < s \text{ for all } 0 \leq i, j \leq n\}.$$

We use $d_X(x_i, x_j) < s$ in this definition because the argument for Proposition 7 does not work for the version of Vietoris–Rips defined with $d_X(x_i, x_j) \leq s$.

► **Definition 3.** A **metric probability space** consists of a metric space X together with a Borel probability measure μ on X .

For a metric space X and $x \in X$, we write $B(x, s)$ for the open ball about x of radius s .

► **Definition 4.** Let (X, μ) be a metric probability space. The **uniform filtration** of (X, μ) is the two-parameter filtration of X where, for $s > 0$ and $k \in [0, 1]$, $X_{(s,k)} \subseteq X$ is the sub-metric space $X_{(s,k)} = \{x \in X : \mu(B(x, s)) \geq k\}$.

The uniform filtration is the special case of the kernel filtration [20, Def. 2.24], where the kernel is chosen to be the uniform kernel [20, Ex. 2.21]. Given a metric probability space, one can take the kernel filtration and then apply any functorial construction on metric spaces. For clustering, a natural choice is single-linkage [20, Def. 2.25]. Applying Vietoris–Rips to the uniform filtration, we get an extension of the usual definition of degree-Rips. This is also considered in Scoccola’s thesis [21, Sec. 6.5]: the stability result mentioned in the introduction is a corollary of a stability result for the kernel filtration.

► **Definition 5.** Let (X, μ) be a metric probability space. The **degree-Rips complex** $\text{DR}(X, \mu)(s, k)$ is the simplicial complex $\text{VR}(X_{(s,k)})(s)$.

We now explain our conventions regarding the circle and annulus. For $R \geq 0$ we write $S_R^1 = \{p \in \mathbb{R}^2 : \|p\| = R\}$, though we sometimes exclude the degenerate case $R = 0$. For $0 \leq R \leq Q$ we write $A_{R,Q} = \{p \in \mathbb{R}^2 : R \leq \|p\| \leq Q\}$. Unless otherwise stated, we view these as metric spaces with the Euclidean metric.

Let $0 < R < Q$, and let $w > 0$. We will consider a metric probability space $\mathcal{A}(R, Q, w)$ that consists of the union of the annulus $A_{R,Q}$ and the inner disc $\{p \in \mathbb{R}^2 : \|p\| < R\}$, with a uniform measure on each piece, such that the measure of the inner disc is equal to w . In more detail, let $\mathcal{A}(R, Q, w)$ be the metric probability space with underlying metric space $\{p \in \mathbb{R}^2 : \|p\| \leq Q\}$, and with probability measure μ given by integrating a density f , where $f(p) = a = w/\pi R^2$ if $\|p\| < R$ and $f(p) = b = (1 - w)/(\pi Q^2 - \pi R^2)$ otherwise. We say that $\mathcal{A}(R, Q, w)$ is a **weighted annulus** if $a < b$. Similarly, let $\mathcal{A}(R, Q, 0)$ be the metric probability space with underlying metric space $A_{R,Q}$, and with probability measure μ given by integrating f , where $f(p) = b = 1/(\pi Q^2 - \pi R^2)$.

3 The Vietoris–Rips complexes of an annulus

In this section we prove that the Vietoris–Rips complexes of an annulus $A_{R,Q}$ are homotopy equivalent to the Vietoris–Rips complexes of the inner circle S_R^1 . This was observed by Adamaszek–Adams [1, Prop. 10.1], who show that it follows from a result of Hausmann [12, Prop. 2.2]. The author overlooked this when writing the first draft of this paper, and it was pointed out by the reviewers.

Proposition 7 below is very similar to Hausmann’s result. If B is a deformation retract of a metric space A , then both results say that the Vietoris–Rips complexes of A and B are homotopy equivalent, provided the deformation retraction is sufficiently compatible with the metric. Proposition 7 assumes slightly less about the deformation retraction than Hausmann’s result, though this is a mild generalization, and may be known to experts. The exposition of the proof here is perhaps more detailed than Hausmann’s, so it remains in the paper, in case it is of interest to some readers.

► **Definition 6.** Let A be a metric space and $B \subseteq A$. We say that B is a **Lipschitz deformation retract** of A if there is a continuous map $r: A \rightarrow B$ such that $r \circ i = \text{id}_B$ where $i: B \hookrightarrow A$ is the inclusion, and there is a homotopy $H: A \times I \rightarrow A(\text{rel } B)$ from id_A to r such that for all $t \in I$, the map $H(-, t): A \rightarrow A$ is 1-Lipschitz.

► **Proposition 7.** If A is a metric space and B is a Lipschitz deformation retract of A , then the inclusion $B \hookrightarrow A$ induces a homotopy equivalence $\text{VR}(B)(s) \simeq \text{VR}(A)(s)$ for all $s > 0$.

► **Remark 8.** Say B is a Lipschitz deformation retract of A , and let $H: A \times I \rightarrow A$ be a homotopy as in the definition. Then for any $t \in I$ and any $s > 0$, there is a simplicial map $H_t^{\text{VR}}: |\text{VR}(A)(s)| \rightarrow |\text{VR}(A)(s)|$ defined by the vertex map $x \mapsto H(x, t)$. These maps appear in the proof of Proposition 7, but note that the function $|\text{VR}(A)(s)| \times I \rightarrow |\text{VR}(A)(s)|$ defined by $(p, t) \mapsto H_t^{\text{VR}}(p)$ is not continuous in general.

► **Lemma 9.** Let K be a simplicial complex, let X be compact, let $f, g: X \rightarrow |K|$ be continuous maps, and let $Z = \{x \in X : f(x) = g(x)\}$. If for all $x \in X$, there is a simplex $\sigma \in K$ with $f(x), g(x) \in |\sigma|$, then f and g are homotopic (rel Z).

Proof. We begin by proving the statement assuming that K is finite. Define the homotopy $H: X \times I \rightarrow |K|$ as follows. For $x \in X$, write $f(x)$ and $g(x)$ in barycentric coordinates, $f(x) = \sum_i \alpha_i v_i$ and $g(x) = \sum_j \beta_j w_j$. For $t \in I$, let $H(x, t) = (1 - t) \cdot \sum_i \alpha_i v_i + t \cdot \sum_j \beta_j w_j$. We have $H(x, t) \in |K|$ since there is $\sigma \in K$ with $f(x), g(x) \in |\sigma|$. We now show that H is continuous. For $\sigma \in K$, let $V_\sigma = f^{-1}(|\sigma|) \cap g^{-1}(|\sigma|)$. As f and g are continuous, V_σ is closed in X , and $\mathcal{V} = \{V_\sigma \times I : \sigma \in K\}$ is a finite, closed cover of $X \times I$. Now $H|_{V_\sigma \times I}: V_\sigma \times I \rightarrow |\sigma|$ is continuous for all σ , and therefore H is continuous.

Now we prove the general statement. As X is compact, there are finite subcomplexes $K_f, K_g \subseteq K$ such that $f(X) \subseteq |K_f|$ and $g(X) \subseteq |K_g|$. Define $L = K_f \cup K_g \cup \{\sigma \cup \tau : \sigma \in K_f, \tau \in K_g, \text{ and } \sigma \cup \tau \in K\}$. Then L is a finite subcomplex of K such that $f(X), g(X) \subseteq |L|$. For $x \in X$, let $\sigma \in K_f$ be the minimal simplex with $f(x) \in |\sigma|$ and let $\tau \in K_g$ be the minimal simplex with $g(x) \in |\tau|$. As $f(x), g(x)$ lie in a common simplex of K , we must have $\sigma \cup \tau \in K$, and thus $f(x), g(x)$ lie in a common simplex of L . Now the statement follows from the finite case. ◀

Proof of Proposition 7. By assumption, there is $r: A \rightarrow B$ such that $r \circ i = \text{id}_B$ where $i: B \hookrightarrow A$ is the inclusion, and there is a homotopy $H: A \times I \rightarrow A(\text{rel } B)$ from id_A to r such that for all $t \in I$, the map $H(-, t): A \rightarrow A$ is 1-Lipschitz. We show that $r_*: |\text{VR}(A)(s)| \rightarrow |\text{VR}(B)(s)|$ induces isomorphisms in π_n for all $n \geq 0$, and the statement follows by Whitehead’s theorem. Since $r \circ i = \text{id}_B$ it follows from functoriality that the induced maps on π_n are surjective for all $n \geq 0$, and it remains to show they are injective.

We begin with π_0 . For $x \in A$, observe that $[x] = [r(x)]$ in $\pi_0 \text{VR}(A)(s)$, as $H(x, -): I \rightarrow A$ is a path from x to $r(x)$. Now, let $x, y \in A$, and say $r_*([x]) = r_*([y])$ in $\pi_0 \text{VR}(B)(s)$. Then in $\pi_0 \text{VR}(A)(s)$, $[x] = [(i \circ r)(x)] = i_*(r_*([x])) = i_*(r_*([y])) = [(i \circ r)(y)] = [y]$.

Now, let $b \in B$ be a choice of basepoint. Since we know r induces an isomorphism on π_0 , it suffices to consider basepoints in B . Say $f: I^n \rightarrow |\text{VR}(A)(s)|$ is a continuous map representing an element of $\pi_n(|\text{VR}(A)(s)|, b)$. As I^n is compact, there is a finite subcomplex $K \subseteq \text{VR}(A)(s)$ such that $f(I^n) \subseteq |K|$. Let $D = \max\{\text{diameter}(\sigma) : \sigma \in K\}$. As K is finite, we have $D < s$. Write $\epsilon = s - D$. For $x \in A$, let $P_x = H(x, -) : I \rightarrow A$. As I is compact, P_x is uniformly continuous, and thus there is $\delta_x > 0$ such that $d_A(P_x(t), P_x(t')) < \epsilon$ when $|t - t'| < \delta_x$. Let $\delta = \min\{\delta_x : x \text{ is a vertex of } K\}$, and choose N such that $1/N < \delta$. For $0 \leq m \leq N$, we write $H_m^{\text{VR}} = H_{\frac{m}{N}}^{\text{VR}}$ for the map induced by H , defined in Remark 8.

We now show that, for any $0 \leq m < N$, $H_m^{\text{VR}} \circ f \simeq H_{m+1}^{\text{VR}} \circ f \text{ (rel } \partial I^n)$. By Lemma 9, it suffices to show that, for all $p \in I^n$, $(H_m^{\text{VR}} \circ f)(p)$ and $(H_{m+1}^{\text{VR}} \circ f)(p)$ lie in a common simplex of $\text{VR}(A)(s)$. For this, choose $\sigma = \{x_0, \dots, x_\ell\} \in K$ with $f(p) \in |\sigma|$; then $\{H_m^{\text{VR}}(x_0), \dots, H_m^{\text{VR}}(x_\ell), H_{m+1}^{\text{VR}}(x_0), \dots, H_{m+1}^{\text{VR}}(x_\ell)\}$ is the desired simplex. To see that it is indeed a simplex of $\text{VR}(A)(s)$, observe that

$$\begin{aligned} d_A(H_m^{\text{VR}}(x_i), H_{m+1}^{\text{VR}}(x_j)) &= d_A(H(x_i, \frac{m}{N}), H(x_j, \frac{m+1}{N})) \\ &\leq d_A(H(x_i, \frac{m}{N}), H(x_i, \frac{m+1}{N})) + d_A(H(x_i, \frac{m+1}{N}), H(x_j, \frac{m+1}{N})) \\ &< \epsilon + d_A(x_i, x_j) \leq s. \end{aligned}$$

It follows that $f \simeq H_N^{\text{VR}} \circ f \text{ (rel } \partial I^n)$. Note that $H_N^{\text{VR}} = i_* \circ r_*$, where $r_*: |\text{VR}(A)(s)| \rightarrow |\text{VR}(B)(s)|$ is as above, and $i_*: |\text{VR}(B)(s)| \rightarrow |\text{VR}(A)(s)|$ is induced by the inclusion $i: B \hookrightarrow A$. Now say that $r_*([f]) = 0$ in $\pi_n(|\text{VR}(B)(s)|, b)$. Then in $\pi_n(|\text{VR}(A)(s)|, b)$, $[f] = [i_* \circ r_* \circ f] = i_*(r_*([f])) = 0$. ◀

► **Corollary 10.** *For any $0 \leq R \leq Q$, the inclusion $S_R^1 \hookrightarrow A_{R,Q}$ induces a homotopy equivalence $\text{VR}(S_R^1)(s) \simeq \text{VR}(A_{R,Q})(s)$ for all $s > 0$.*

Proof. The homotopy $H: A_{R,Q} \times I \rightarrow A_{R,Q}$ defined by $((r, \theta), t) \mapsto ((1-t) \cdot r + t \cdot R, \theta)$ shows that S_R^1 is a Lipschitz deformation retract of $A_{R,Q}$. ◀

4 Boundary curves in the degree-Rips parameter space

In this section we prove Theorem 1. To motivate the approach, we first explain the basic idea for how to compute the homotopy type of $\text{DR}(\mathcal{A}(R, Q, w))(s, k)$ for a particular choice of s and k . We will show in this section that the subspace $\mathcal{A}(R, Q, w)_{(s,k)}$ is an annulus; write P for its inner radius. By Corollary 10, $\text{DR}(\mathcal{A}(R, Q, w))(s, k) \simeq \text{VR}(S_P^1)(s)$, and then one concludes by the Adamaszek–Adams calculation of the Vietoris–Rips complexes of the circle.

To begin, we need to compute the measure of an s -ball in $\mathcal{A}(R, Q, w)$, and for this we need to know the area of the intersection of the s -ball with the annulus, and with the inner disc. We now briefly explain how to do this, though we omit most formulas for brevity.

Let $R > 0$ and $s > 0$. We write O for the origin in \mathbb{R}^2 and $C(p, r)$ for the circle centred at p of radius r . Define a function $\alpha: [0, \infty) \rightarrow \mathbb{R}$ by letting $\alpha(c)$ be the area of the intersection $B(O, R) \cap B((c, 0), s)$. We calculate $\alpha(c)$ using the usual formulas for the area of circular segments, but there are several cases. In Figure 3 on the left we show the (c, s) -space, with the curves that delimit the cases:

$$c + s = R \tag{1}$$

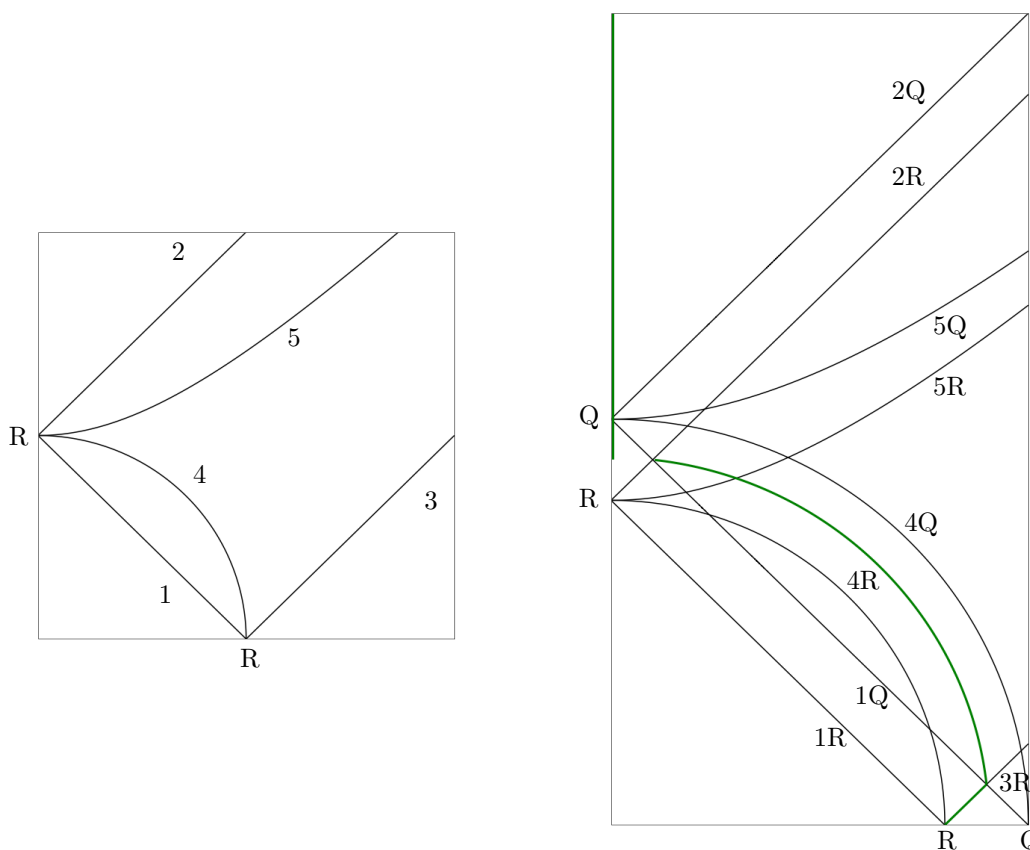
$$s - c = R \tag{2}$$

$$c - s = R \tag{3}$$

$$c^2 + s^2 = R^2 \tag{4}$$

$$s^2 - c^2 = R^2 \tag{5}$$

If (c, s) is outside the region bounded by the lines 1, 2, 3, then the circles $C(O, R)$ and $C((c, 0), s)$ do not intersect, and α is constant. If (c, s) is inside this region, then the circles intersect in two points: if (c, s) is to the left of the circle 4, then the centre $(c, 0)$ is to the left of the chord connecting these points of intersection, otherwise it is to the right; if (c, s) is to the left of the hyperbola 5, then the origin is to the right of this chord, otherwise it is to the left. In each region, we calculate $\alpha(c)$ as a sum of areas of circular segments, or their complements.



■ **Figure 3** Let $R > 0$. We can think of α_R as a function of two variables c and s . On the left, we show the domain of α_R , with c on the horizontal axis and s on the vertical axis. In each region we compute α_R using a sum of areas of circular segments, or their complements. To compute γ , we need to compute α_R and α_Q . On the right we plot the domain of γ as a function of c and s , with $R = 0.4$ and $Q = 0.5$. In green, we plot $\omega(s)$ with $w = 0$. In this case $\omega(s)$ is the value of c that maximizes the area of the intersection between the annulus $A_{R,Q}$ and the s -ball centred at $(c, 0)$.

In order to prove the main theorem, we need to understand how the area of a ball in $\mathcal{A}(R, Q, w)$ changes as its centre varies. For this it will be helpful to understand the derivative of α . This seems complicated to compute directly from the formula for the area of a circular segment, but instead we can compute the derivative using a geometric argument. Define $y: [0, \infty) \rightarrow \mathbb{R}$ by

$$y(c) = \begin{cases} 0 & \text{if } C(O, R) \cap C((c, 0), s) = \emptyset \\ \max\{x_2 : (x_1, x_2) \in C(O, R) \cap C((c, 0), s)\} & \text{else} \end{cases}$$

If the circles intersect in exactly two points, then $y(c)$ is simply the y -coordinate of these points, with positive sign, and then $2y(c)$ is the length of the chord connecting these points of intersection.

► **Lemma 11.** *The map α is continuously differentiable and $\alpha'(c) = -2y(c)$.*

Proof. Since α can be computed by summing the areas of circular segments, the formula for this area shows that α is differentiable. By definition,

$$\alpha'(c) = \lim_{\delta \rightarrow 0} \frac{\alpha(c + \delta) - \alpha(c)}{\delta}.$$

Now, $\alpha(c + \delta) - \alpha(c)$ is the (signed) area of the subset of $B(O, R)$ between the circles $C((c, 0), s)$ and $C((c + \delta, 0), s)$. And, $-2y(c) \cdot \delta$ approximates this area, in the sense that we have the following inequality for all s, c and small enough δ :

$$|(\alpha(c + \delta) - \alpha(c)) + 2y(c) \cdot \delta| \leq 2 \cdot |\delta| \cdot |y(c) - y(c + \delta)|.$$

So,

$$\begin{aligned} |\alpha'(c) + 2y(c)| &\leq \lim_{\delta \rightarrow 0} \frac{2 \cdot |\delta| \cdot |y(c) - y(c + \delta)|}{|\delta|} \\ &= \lim_{\delta \rightarrow 0} 2 \cdot |y(c) - y(c + \delta)| = 0. \end{aligned} \quad \blacktriangleleft$$

Let $0 < R < Q$. We now consider the area of the intersection of the ball $B((c, 0), s)$ with the annulus $A_{R,Q} = \{p \in \mathbb{R}^2 : R \leq \|p\| \leq Q\}$. Define a function $\gamma: [0, Q] \rightarrow \mathbb{R}$ by letting $\gamma(c)$ be the area of the intersection $A_{R,Q} \cap B((c, 0), s)$. We change notation in order to relate γ with α : instead of α we write α_R , where R is the radius of the circle centred at the origin. With this notation, $\gamma = \alpha_Q - \alpha_R$. We calculate γ using this formula, though now there are more cases; see Figure 3 on the right for the curves delimiting the cases.

Again we want to understand γ' . The interesting case is when the ball $B((c, 0), s)$ intersects both circles $C(O, R)$ and $C(O, Q)$. This happens when (c, s) is in the region bounded by the lines:

$$c + s = Q \tag{1Q}$$

$$s - c = R \tag{2R}$$

$$c - s = R \tag{3R}$$

We now collect together the facts we will need about γ :

► **Lemma 12.** *Say $c + s > Q$ and $s - c < R$ and $c - s < R$. Then,*

- $\gamma'(c) = 0$ if and only if $c^2 + s^2 = \frac{1}{2}(R^2 + Q^2)$
- $\gamma'(c) > 0$ if and only if $c^2 + s^2 < \frac{1}{2}(R^2 + Q^2)$
- $\gamma'(c) < 0$ if and only if $c^2 + s^2 > \frac{1}{2}(R^2 + Q^2)$.

Moreover, if we have $c \leq c'$ with $c' + s > Q$ and $s - c' < R$ and $c' - s < R$ and $\gamma'(c), \gamma'(c') > 0$, then $\gamma'(c) \geq \gamma'(c')$.

Proof. As $\gamma'(c) = \alpha'_Q(c) - \alpha'_R(c)$, we have $\gamma'(c) = -2y_Q(c) + 2y_R(c)$ by Lemma 11. So, $\gamma'(c) = 0$ if and only if $y_Q(c) = y_R(c)$; and $\gamma'(c) > 0$ if and only if $y_Q(c) < y_R(c)$; and $\gamma'(c) < 0$ if and only if $y_Q(c) > y_R(c)$.

Now, $C(O, R)$ and $C((c, 0), s)$ intersect in two points, and the x -coordinate of both points is given by $x_R(c) = (c^2 + R^2 - s^2)/2c$, and then $y_R(c) = \sqrt{R^2 - x_R(c)^2}$. Similarly we compute $y_Q(c)$. Now, a little algebra shows that $y_Q(c) = y_R(c)$ if and only if $c^2 + s^2 = \frac{1}{2}(R^2 + Q^2)$; and $y_Q(c) < y_R(c)$ if and only if $c^2 + s^2 < \frac{1}{2}(R^2 + Q^2)$; and $y_Q(c) > y_R(c)$ if and only if $c^2 + s^2 > \frac{1}{2}(R^2 + Q^2)$.

To prove the last statement of the lemma, we must show that $y_R(c) - y_Q(c) \geq y_R(c') - y_Q(c')$. Towards a contradiction, assume $y_R(c) - y_Q(c) < y_R(c') - y_Q(c')$. Then a little more algebra shows that $c'^2(R^2 + Q^2 - 2s^2) < c^2(R^2 + Q^2 - 2s^2)$. As we assume $\gamma'(c) > 0$, we have $c^2 + s^2 < \frac{1}{2}(R^2 + Q^2)$, and therefore $s^2 < \frac{1}{2}(R^2 + Q^2)$. So, we conclude $c' < c$, a contradiction. \blacktriangleleft

We are ready to show that the subspace $\mathcal{A}(R, Q, w)_{(s,k)}$ is an annulus, for any choice of s and k . This will follow from the next lemma.

Now, for $s > 0$, let $\nu: [0, Q] \rightarrow [0, 1]$ be the function $\nu(c) = \mu(B((c, 0), s))$, where μ is the measure on $\mathcal{A}(R, Q, w)$. We write ν_s if it is necessary to specify s . Since ν is a linear combination of γ and α_R , we have already seen how to calculate ν . Define $\omega: (0, \infty) \rightarrow [0, Q]$ as follows. For any $s > 0$, let $M_s = \max_{c \in [0, Q]} \nu_s(c)$. As ν_s is continuous, $\nu_s^{-1}(M_s) \subseteq [0, Q]$ is non-empty and closed, and we let $\omega(s) = \min(\nu_s^{-1}(M_s))$.

► **Lemma 13.** *For any $s > 0$, ν_s is non-decreasing on $[0, \omega(s)]$ and non-increasing on $[\omega(s), Q]$.*

Proof. There are three cases: (1) $0 < s \leq \frac{1}{2}(Q - R)$; (2) $\frac{1}{2}(Q - R) < s < \frac{1}{2}(Q + R)$; and (3) $\frac{1}{2}(Q + R) \leq s$. See Figure 3 for an idea of how ω behaves in the three cases.

Case (1) is straightforward. On $[0, R - s]$ ν is constant, it is strictly increasing on $[R - s, R + s]$, and we have $\nu(c) = M_s$ for any $c \in [R + s, Q - s]$; so $\omega(s) = R + s$. On $[Q - s, Q]$ ν is strictly decreasing.

We now consider Case (2). On $[0, R - s]$ ν is constant. In the region bounded by lines $1R, 1Q, 2R$, ν is strictly increasing. If c is to the left of line $2R$, then ν is constant. The interesting case is that c is to the right of line $1Q$: $c \geq Q - s$. We will show that $\omega(s)$ is in this region, and that $\nu' > 0$ on $[Q - s, \omega(s))$, $\nu'(\omega(s)) = 0$, and $\nu' < 0$ on $(\omega(s), Q]$.

Let $z = \sqrt{\frac{1}{2}(R^2 + Q^2) - s^2}$. Then by Lemma 12, $\gamma'(c) < 0$ for all $c \in (z, Q]$, and $\alpha'_R(c) \leq 0$, so $\nu'(c) < 0$. When $c = Q - s$, so that c is on line $1Q$, we have already seen that $\nu'(c) \geq 0$. So, as ν' is continuous, it must have a zero on $[Q - s, z]$. We now show that ν' has at most one zero on this interval, and it follows that $\omega(s)$ is this zero.

Let $c \leq c'$ in $[Q - s, z]$ such that $\nu'(c) = \nu'(c') = 0$. As $\nu = a \cdot \alpha_R + b \cdot \gamma$, $\nu'(c) = 0$ implies that $\gamma'(c) = -\frac{a}{b} \cdot \alpha'_R(c)$. We have $\gamma = \alpha_Q - \alpha_R$, so $\nu'(c) = 0$ implies that $(1 - \frac{a}{b}) \cdot \alpha'_R(c) = \alpha'_Q(c)$. By Lemma 12, since $c \leq c'$ in $[Q - s, z]$, we have $\gamma'(c) \geq \gamma'(c')$. So,

$$\begin{aligned} -\frac{a}{b} \cdot \alpha'_R(c) &\geq -\frac{a}{b} \cdot \alpha'_R(c') \\ (1 - \frac{a}{b}) \cdot \alpha'_R(c) &\leq (1 - \frac{a}{b}) \cdot \alpha'_R(c') \\ \alpha'_Q(c) &\leq \alpha'_Q(c') \\ y_Q(c) &\geq y_Q(c'). \end{aligned}$$

As y_Q is strictly increasing on $[Q - s, \sqrt{Q^2 - s^2}]$ we have $c = c'$. This finishes Case (2).

Case (3) is straightforward; $\omega(s) = 0$. If c is to the left of line $2Q$, or c is in the region bounded by lines $1Q$ and $2R$, then ν is constant. If c is in the region bounded by lines $1Q, 2R$ and $2Q$, then ν is strictly decreasing. If c is to the right of line $2R$ then $\nu'(c) < 0$ by Lemma 12. \blacktriangleleft

Note that this proof also shows how to compute $\omega(s)$. If $0 < s \leq \frac{1}{2}(Q - R)$, then $\omega(s) = R + s$. If $\frac{1}{2}(Q - R) < s < \frac{1}{2}(Q + R)$, then $\omega(s)$ is defined implicitly by the equation $(b - a) \cdot y_R(c) = b \cdot y_Q(c)$. In this case, if $w = 0$, then this last equation simplifies to $c^2 + s^2 = \frac{1}{2}(R^2 + Q^2)$. If $\frac{1}{2}(Q + R) \leq s$, $\omega(s) = 0$.

We are almost ready to define the maps φ_ℓ and prove Theorem 1. In order to make use of the Adamaszek–Adams calculation of the Vietoris–Rips complexes of the circle, we need to relate the Vietoris–Rips complexes of a circle with the Euclidean distance to the Vietoris–Rips complexes of the circle with geodesic distance. For this we use the following lemma, whose proof is straightforward.

► **Lemma 14.** *Let (X_1, d_1) and (X_2, d_2) be metric spaces, let $D_1 = \text{Im}(d_1) \subseteq \mathbb{R}_{\geq 0}$ and $D_2 = \text{Im}(d_2) \subseteq \mathbb{R}_{\geq 0}$, and say there is a bijection $f: X_1 \rightarrow X_2$ and an order-preserving bijection $f_d: D_1 \rightarrow D_2$ such that $f_d \circ d_1 = d_2 \circ (f \times f)$. Then f induces an isomorphism $\text{VR}(X_1, d_1)(s) \cong \text{VR}(X_2, d_2)(f_d(s))$ for any $s \in D_1$.*

For $r > 0$, we write (S_r^1, d_g) for the circle of radius r equipped with the geodesic metric, and (S_r^1, d_E) for the circle of radius r equipped with the Euclidean metric. Define $\sigma_r: [0, 2r] \rightarrow [0, \pi r]$ by $\sigma_r(t) = 2r \arcsin(\frac{t}{2r})$. If $p, q \in S_r^1$, then $\sigma_r(d_E(p, q)) = d_g(p, q)$.

By Adamaszek–Adams [1, Theorem 7.4],

$$\text{VR}(S_{\frac{1}{2\pi}}^1, d_g)(s) \simeq S^{2\ell+1} \quad \text{for} \quad \frac{\ell}{2\ell+1} < s \leq \frac{\ell+1}{2\ell+3}, \ell = 0, 1, \dots$$

And, if $\frac{\ell}{2\ell+1} < s \leq s' \leq \frac{\ell+1}{2\ell+3}$, then the inclusion $\text{VR}(S_{\frac{1}{2\pi}}^1, d_g)(s) \hookrightarrow \text{VR}(S_{\frac{1}{2\pi}}^1, d_g)(s')$ is a homotopy equivalence.

In order to define the maps φ_ℓ , we need to find, for any Vietoris–Rips parameter value $s > 0$, the radius r such that $\text{VR}(S_r^1, d_E)(s)$ is isomorphic to $\text{VR}(S_{\frac{1}{2\pi}}^1, d_g)(\frac{\ell}{2\ell+1})$. So, for any integer $\ell > 0$, let $\rho_\ell: (0, \infty) \rightarrow (0, \infty)$ be defined by

$$\rho_\ell(s) = \frac{s}{2 \sin(\frac{\pi \ell}{2\ell+1})}.$$

Then, for any $s > 0$ we have

$$\frac{\ell}{2\ell+1} = \frac{\sigma_{\rho_\ell(s)}(s)}{2\pi \rho_\ell(s)},$$

and therefore by Lemma 14, we have

$$\text{VR}(S_{\rho_\ell(s)}^1, d_E)(s) \cong \text{VR}(S_{\frac{1}{2\pi}}^1, d_g)(\frac{\ell}{2\ell+1}).$$

Similarly, define $\rho_\infty: (0, \infty) \rightarrow (0, \infty)$ by $\rho_\infty(s) = s/2$.

We can now define the maps $\varphi_\ell: (0, \infty) \rightarrow [0, 1]$ for $\ell = 0, 1, 2, \dots, \infty$. For the case $\ell = 0$, we let $\varphi_0(s) = \nu_s(\omega(s)) = M_s$. For $\ell > 0$, let

$$\varphi_\ell(s) = \nu_s(\min(\rho_\ell(s), \omega(s))).$$

Note that, by Lemma 13, for any $s > 0$ and $0 \leq \ell < \ell' \leq \infty$, we have $\varphi_\ell(s) \geq \varphi_{\ell'}(s)$.

Proof of Theorem 1. Write $\mathcal{A} = \mathcal{A}(R, Q, w)$. If $k > \varphi_0(s) = M_s$, then $\mathcal{A}_{(s,k)} = \emptyset$, so that $\text{DR}(\mathcal{A})(s, k) = \emptyset$. Next, we show that if $\mathcal{A}_{(s,k)}$ is non-empty, then it is an annulus. Now,

$$\begin{aligned} \mathcal{A}_{(s,k)} &= \{p \in \mathcal{A} \mid \mu(B(p, s)) \geq k\} \\ &= (\nu_s \circ \|\cdot\|)^{-1}([k, 1]) \end{aligned}$$

which is closed as ν_s and $\|\cdot\|$ are continuous. It suffices to show that $\nu_s^{-1}([k, 1]) \subset [0, Q]$ is an interval, and this follows from Lemma 13.

58:12 The Degree-Rips Complexes of an Annulus with Outliers

Now, say that $0 < \ell < \infty$ and $s > 0$ and $k \in [0, 1]$ are such that $\varphi_\ell(s) > k > \varphi_{\ell+1}(s)$. Let P be the left endpoint of the interval $\nu_s^{-1}([k, 1])$, so that $\mathcal{A}_{(s,k)}$ is an annulus with inner radius P . We show now that $\rho_{\ell+1}(s) < P < \rho_\ell(s)$.

As $\varphi_\ell(s) \neq \varphi_{\ell+1}(s)$ and $\rho_\ell(s) > \rho_{\ell+1}(s)$, we have $\rho_{\ell+1}(s) < \omega(s)$ and $\varphi_{\ell+1}(s) = \nu_s(\rho_{\ell+1}(s))$. As $k > \varphi_{\ell+1}(s) = \nu_s(\rho_{\ell+1}(s))$ we have $\rho_{\ell+1}(s) \notin \nu_s^{-1}([k, 1])$; as $\omega(s) \in \nu_s^{-1}([k, 1])$ and $\rho_{\ell+1}(s) < \omega(s)$, we have $\rho_{\ell+1}(s) < P$, as desired. By continuity of ν_s , there is $r \in (\rho_{\ell+1}(s), P]$ with $\nu_s(r) = k$. By definition of P , we have $P \leq r$, and thus $P = r$ and $\nu_s(P) = k$. Since $\varphi_\ell(s) > k$, we have $P < \rho_\ell(s)$.

Now, by Corollary 10, the inclusion $S_P^1 \hookrightarrow \mathcal{A}_{(s,k)}$ induces a homotopy equivalence $\text{VR}(S_P^1)(s) \simeq \text{VR}(\mathcal{A}_{(s,k)})(s)$. By Lemma 14, $\text{VR}(S_P^1)(s) \cong \text{VR}(S_{\frac{1}{2\pi}}^1, d_g)(\frac{\sigma_P(s)}{2\pi P})$.

As $\rho_{\ell+1}(s) < P < \rho_\ell(s)$, we have

$$\frac{\ell+1}{2\ell+3} = \frac{\sigma_{\rho_{\ell+1}(s)}(s)}{2\pi\rho_{\ell+1}(s)} > \frac{\sigma_P(s)}{2\pi P} > \frac{\sigma_{\rho_\ell(s)}(s)}{2\pi\rho_\ell(s)} = \frac{\ell}{2\ell+1}$$

So that $\text{DR}(\mathcal{A})(s, k) = \text{VR}(\mathcal{A}_{(s,k)})(s) \simeq \text{VR}(S_P^1)(s) \simeq S^{2\ell+1}$ by [1, Theorem 7.4].

If s and k are such that $\varphi_\infty(s) > k$, then we have seen that $\mathcal{A}_{(s,k)}$ is an annulus, and again we write P for the inner radius. Then one checks that $P < \rho_\infty(s) = s/2$, and so $\text{VR}(S_P^1)(s)$ is contractible.

The claim that inclusions $\text{DR}(\mathcal{A})(s, k) \hookrightarrow \text{DR}(\mathcal{A})(s', k')$ are homotopy equivalences whenever (s, k) and (s', k') both lie between φ_ℓ and $\varphi_{\ell+1}$ follows from Corollary 10 and the statement in [1, Theorem 7.4] about inclusions of Vietoris–Rips complexes. \blacktriangleleft

5 The annulus without outliers

We now consider the case $w = 0$, when the measure of the inner disc $\{p \in \mathbb{R}^2 : \|p\| < R\}$ is zero. The measure of an s -ball $B(p, s)$ is not much changed from the case where w is small but non-zero. However, the degree-Rips complexes of $\mathcal{A}(R, Q, 0)$ exhibit different behavior from the case $w > 0$, because now the vertices of the degree-Rips complexes must lie in the annulus $A_{R,Q}$. In this section, we modify the constructions of Section 4 accordingly, and then prove the analogue of Theorem 1 in this case.

For $s > 0$, let $\tilde{\nu}_s: [R, Q] \rightarrow [0, 1]$ be defined by $\tilde{\nu}_s(c) = \mu(B((c, 0), s))$. Define $\tilde{\omega}: (0, \infty) \rightarrow [R, Q]$ as follows. For any $s > 0$, let $M_s = \max_{c \in [R, Q]} \tilde{\nu}_s(c)$. As $\tilde{\nu}_s$ is continuous, $\tilde{\nu}_s^{-1}(M_s) \subseteq [R, Q]$ is non-empty and closed, and we let $\tilde{\omega}(s) = \min(\tilde{\nu}_s^{-1}(M_s))$.

As before, $\tilde{\varphi}_0: (0, \infty) \rightarrow [0, 1]$ is defined as $\tilde{\varphi}_0(s) = M_s = \tilde{\nu}_s(\tilde{\omega}(s))$. But now, for $0 < \ell \leq \infty$, we define $\tilde{\varphi}_\ell: (0, \infty) \rightarrow [0, 1]$ by

$$\tilde{\varphi}_\ell(s) = \begin{cases} 0 & \text{if } \rho_\ell(s) \leq R \\ \tilde{\nu}_s(\min(\rho_\ell(s), \tilde{\omega}(s))) & \text{else} \end{cases}$$

Note that the $\tilde{\varphi}_\ell$ need not be continuous.

► **Theorem 15.** *For any $s > 0$ and any $k \in [0, 1]$,*

$$\text{DR}(\mathcal{A}(R, Q, 0))(s, k) \simeq \begin{cases} \emptyset & \text{if } k > \tilde{\varphi}_0(s) \\ S^{2\ell+1} & \text{if } \tilde{\varphi}_\ell(s) > k > \tilde{\varphi}_{\ell+1}(s) \text{ for } \ell \neq \infty \\ * & \text{if } \tilde{\varphi}_\infty(s) > k \end{cases}$$

Moreover, if $0 < \ell < \infty$ and $0 < s \leq s'$ and $0 \leq k' \leq k \leq 1$ are such that

$$\tilde{\varphi}_\ell(s) > k > \tilde{\varphi}_{\ell+1}(s) \quad \text{and} \quad \tilde{\varphi}_\ell(s') > k' > \tilde{\varphi}_{\ell+1}(s'),$$

then the inclusion $\text{DR}(\mathcal{A}(R, Q, 0))(s, k) \hookrightarrow \text{DR}(\mathcal{A}(R, Q, 0))(s', k')$ is a homotopy equivalence.

Proof. The proof is quite similar to the proof of Theorem 1. If $s > 0$ and $k \in [0, 1]$ are such that $\tilde{\varphi}_\ell(s) > k > \tilde{\varphi}_{\ell+1}(s)$, then, arguing as before, $\mathcal{A}(R, Q, 0)_{(s,k)}$ is an annulus with inner radius P such that $\rho_{\ell+1}(s) \leq P < \rho_\ell(s)$. Therefore,

$$\frac{\ell+1}{2\ell+3} = \frac{\sigma_{\rho_{\ell+1}(s)}(s)}{2\pi\rho_{\ell+1}(s)} \geq \frac{\sigma_P(s)}{2\pi P} > \frac{\sigma_{\rho_\ell(s)}(s)}{2\pi\rho_\ell(s)} = \frac{\ell}{2\ell+1}$$

So that

$$\text{DR}(\mathcal{A}(R, Q, 0))_{(s,k)} = \text{VR}(\mathcal{A}(R, Q, 0)_{(s,k)})(s) \simeq \text{VR}(S_P^1)(s) \simeq S^{2\ell+1}$$

again by [1, Theorem 7.4]. The claim about inclusions of degree-Rips complexes is proved in the same way as before. ◀

6 Conclusions

In various experiments, and in this paper, we have observed the following behavior. If there is a strong topological signal in data, and this appears somewhere in the parameter space of degree-Rips, then if one adds outliers, the topological signal is still visible (i.e., prominent) in degree-Rips, but at a different location in the parameter space, where the values of the Rips parameter are smaller.

The main interest of the calculation in this paper is that, in this case, it is possible to say precisely how the location of the signal changes in the degree-Rips parameter space. We now briefly mention one reason why we would like to understand this in more general settings. If one is interested in taking one-parameter slices of degree-Rips (e.g., for computing a barcode, or for clustering as in robust single-linkage [8] or γ -linkage [20]), then choosing the slice is tricky in practice. But it seems that, both for computational reasons and to maximize robustness, one wants to choose a slice through “small” values of the Rips parameter. A satisfactory understanding of the robustness of degree-Rips may shed light on this.

There are several directions in which one could try to extend the results of this paper. Of course it would be interesting to consider measures supported on more complicated spaces, perhaps seeking only partial calculations or approximations. One could also consider other models for outliers. For example, one could take a convolution with a kernel (as in [19, Section 2.1]), rather than mixing with a uniform measure. Finally, a reviewer posed the following question: is there a density on the disc that is rotationally invariant and monotone in the radius such that the uniform filtration at some parameter (s, k) is not an annulus?

References

- 1 Michał Adamaszek and Henry Adams. The Vietoris–Rips complexes of a circle. *Pac. J. Math.*, 290(1):1–40, 2017. doi:10.2140/pjm.2017.290.1.
- 2 Dominique Attali, André Lieutier, and David Salinas. Vietoris–Rips complexes also provide topologically correct reconstructions of sampled shapes. *Computational Geometry*, 46(4):448–465, 2013. 27th Annual Symposium on Computational Geometry (SoCG 2011). doi:10.1016/j.comgeo.2012.02.009.
- 3 Paul Bendich, David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Dmitriy Morozov. Inferring local homology from sampled stratified spaces. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS2007), October 20–23, 2007, Providence, RI, USA, Proceedings*, pages 536–546. IEEE Computer Society, 2007. doi:10.1109/FOCS.2007.33.
- 4 Andrew J. Blumberg and Michael Lesnick. Stability of 2-parameter persistent homology, 2020. arXiv:2010.09628.

- 5 Omer Bobrowski, Sayan Mukherjee, and Jonathan E. Taylor. Topological consistency via kernel estimation. *Bernoulli*, 23(1):288–328, 2017. doi:10.3150/15-BEJ744.
- 6 Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and Topological Inference*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2018. doi:10.1017/9781108297806.
- 7 Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, volume 7819 of *Lecture Notes in Computer Science*, pages 160–172. Springer, 2013.
- 8 Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for the cluster tree. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 343–351. Curran Associates, Inc., 2010.
- 9 Frédéric Chazal, Leonidas J. Guibas, Steve Y. Oudot, and Primoz Skraba. Persistence-based clustering in Riemannian manifolds. *J. ACM*, 60(6), November 2013. doi:10.1145/2535927.
- 10 Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. Estimating the number of clusters. *Canadian Journal of Statistics*, 28:367–382, 2000.
- 11 Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.
- 12 Jean-Claude Hausmann. On the Vietoris–Rips complexes and a cohomology theory for metric spaces. In *Prospects in topology. Proceedings of a conference in honor of William Browder, Princeton, NJ, USA, March 1994*, pages 175–188. Princeton, NJ: Princeton University Press, 1995.
- 13 J. F. Jardine. Stable components and layers. *Canad. Math. Bull.*, pages 1–15, 2019. doi:10.4153/S000843951900064X.
- 14 J. F. Jardine. Persistent homotopy theory, 2020. arXiv:2002.10013.
- 15 Michael Lesnick and Matthew Wright. Interactive visualization of 2-D persistence modules, 2015. arXiv:1512.00180.
- 16 Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, volume 00, pages 33–42, November 2018. doi:10.1109/ICDMW.2017.12.
- 17 Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete Comput. Geom.*, 39:419–441, 2008. doi:10.1007/s00454-008-9053-2.
- 18 Steve Y. Oudot and Donald R. Sheehy. Zigzag zoology: Rips zigzags for homology inference. *Found Comput Math*, 15(5):1151–1186, 2015. doi:10.1007/s10208-014-9219-7.
- 19 Alessandro Rinaldo and Larry Wasserman. Generalized density clustering. *Ann. Statist.*, 38(5):2678–2722, October 2010. doi:10.1214/10-AOS797.
- 20 Alexander Rolle and Luis Scoccola. Stable and consistent density-based clustering, 2020. arXiv:2005.09048.
- 21 Luis Scoccola. Locally persistent categories and metric properties of interleaving distances. *Electronic Thesis and Dissertation Repository*. <https://ir.lib.uwo.ca/etd/7119>, 2020.
- 22 The RIVET Developers. RIVET. 1.1.0, 2020. URL: <https://github.com/rivetTDA/rivet/>.

Chains, Koch Chains, and Point Sets with Many Triangulations

Daniel Rutschmann ✉

Department of Computer Science, ETH Zürich, Switzerland

Manuel Wettstein¹ ✉

Department of Computer Science, ETH Zürich, Switzerland

Abstract

We introduce the abstract notion of a chain, which is a sequence of n points in the plane, ordered by x -coordinates, so that the edge between any two consecutive points is unavoidable as far as triangulations are concerned. A general theory of the structural properties of chains is developed, alongside a general understanding of their number of triangulations.

We also describe an intriguing new and concrete configuration, which we call the Koch chain due to its similarities to the Koch curve. A specific construction based on Koch chains is then shown to have $\Omega(9.08^n)$ triangulations. This is a significant improvement over the previous and long-standing lower bound of $\Omega(8.65^n)$ for the maximum number of triangulations of planar point sets.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Planar Point Set, Chain, Koch Chain, Triangulation, Maximum Number of Triangulations, Lower Bound

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.59

Related Version *Full Version:* <http://arxiv.org/abs/2203.07584>

Acknowledgements The material presented in this paper originates from the first author's Master's thesis [18] under the second author's direct supervision. Both authors wish to express their gratitude to Emo Welzl, the official advisor in this endeavor.

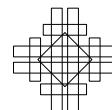
1 Introduction

Let P be a set of n points in the Euclidean plane. Throughout the paper, P is assumed to be in *general position*, which means for us that no two points have the same x -coordinate and that no three points are on a common line. A *geometric graph* on P is a graph with vertex set P combined with an embedding into the plane where edges are realized as straight-line segments between the corresponding endpoints. It is called *crossing-free* if the edges have no pairwise intersection, except possibly in a common endpoint.

Triangulations. Perhaps the most prominent and most studied family of crossing-free geometric graphs is the family of *triangulations*, which may be defined simply as edge-maximal crossing-free geometric graphs on P . It is easy to see that such a definition implies that the edges of any triangulation subdivide the convex hull of P into triangular regions.

Let $\text{tr}(P)$ denote the number of triangulations on a given point set P . Trying to better understand this quantity is a fundamental question in combinatorial and computational geometry. For very specific families of point sets, exact formulas or at least asymptotic

¹ Corresponding author



estimates can be derived. For example, it is well-known that if P is in *convex position*, then $\text{tr}(P) = C_{n-2}$ where $C_k = \frac{1}{k+1} \binom{2k}{k} = \Theta(k^{-3/2}4^k)$ is the k -th Catalan number [1]. In general, however, this problem turns out to be much more elusive.

There is an elegant algorithm by Alvarez and Seidel [7] from 2013 that computes $\text{tr}(P)$ in exponential time $O(2^n n^2)$. It was surpassed by Marx and Miltzow [16] in 2016, who showed how to compute $\text{tr}(P)$ in subexponential time $n^{O(\sqrt{n})}$. Moreover, Avis and Fukuda [9] have shown already in 1996 how to enumerate the set of all triangulations on P (i.e., to compute an explicit representation of each element) by using a general technique called reverse search in time $\text{tr}(P) \cdot p(n)$ for some polynomial p . A particularly efficient implementation of that technique with $p(n) = O(\log \log n)$ has been described by Bepamyatnikh [10].

Extensive research has also gone into extremal upper and lower bounds in terms of the number of points. That is, if we define

$$\text{tr}_{\max}(n) = \max_{P: |P|=n} \text{tr}(P), \quad \text{tr}_{\min}(n) = \min_{P: |P|=n} \text{tr}(P)$$

to be the respectively largest and smallest numbers of triangulations attainable by a set P of n points in general position, then various authors have attempted to establish and improve upper and lower bounds on these quantities.

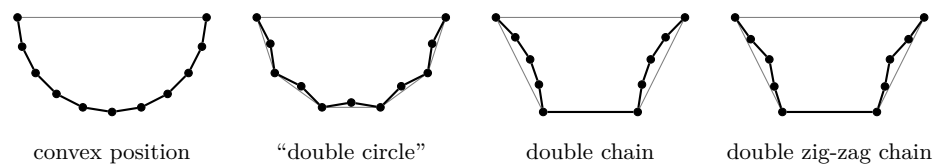
As far as the maximum is concerned, a seminal result by Ajtai, Chvátal, Newborn, and Szemerédi [6] from 1982 shows that the number of triangulations – and, more generally, the number of all crossing-free geometric graphs – is at most 10^{13n} . A long series of successive improvements [23, 11, 20, 19, 22] using a variety of different techniques has culminated in the currently best upper bound $\text{tr}_{\max}(n) \leq 30^n$ due to Sharir and Sheffer [21], which has remained uncontested for over a decade. Coming from the other side, attempts have been made to construct point sets with a particularly large number of triangulations. For some time, the *double chain* by García, Noy, and Tejel [17] with approximately $\Theta(8^n)$ triangulations was conjectured to have the largest possible number of triangulations. However, variants like the *double zig-zag chain* by Aichholzer et al. [5] with $\Theta(8.48^n)$ triangulations and a specific instance of the *generalized double zig-zag chain* by Dumitrescu, Schulz, Sheffer, and Tóth [12] with $\Omega(8.65^n)$ triangulations have since been discovered. But also on this front, no further progress on the lower bound $\text{tr}_{\max}(n) = \Omega(8.65^n)$ has been made for a decade.

The situation for the minimum is different insofar that the *double circle* with $\Theta(3.47^n)$ triangulations, as analyzed by Hurtado and Noy [15] in 1997, is still conjectured by many to have the smallest number of triangulations. In other words, it is believed that the resulting upper bound $\text{tr}_{\min}(n) = O(3.47^n)$ is best possible. On the other hand, Aichholzer et al. [4] have shown that every point set has at least $\Omega(2.63^n)$ triangulations, thereby establishing the lower bound $\text{tr}_{\min}(n) = \Omega(2.63^n)$.

The focus of this paper lies on $\text{tr}_{\max}(n)$ and, more specifically, on establishing an improved lower bound on that quantity. Ultimately, we show how to construct a new infinite family of point sets with $\Omega(9.08^n)$ triangulations, thereby proving $\text{tr}_{\max}(n) = \Omega(9.08^n)$.

General chains. It has occurred to us that almost all families of point sets whose numbers of triangulations have been analyzed over the years have a very special structure, which we are trying to capture in the following definition.

► **Definition 1.** A chain C is a sequence of points p_0, \dots, p_n sorted by increasing x -coordinates, such that the edge $p_{i-1}p_i$ is unavoidable (i.e., contained in every triangulation of C) for each $i = 1, \dots, n$. These specific unavoidable edges are also referred to as chain edges.



■ **Figure 1** Some classic point sets realized as chains. For the double circle, we need to remove one of the inner points. Chain edges are displayed black and bold, other unavoidable hull edges in gray.

In contrast to previous convention, we use the parameter n to denote the number of chain edges and not the number of points in C , which is $n + 1$. Also note that Definition 1 implies that the edge p_0p_n is an edge of the convex hull and, hence, also unavoidable. Indeed, since all chain edges are unavoidable, the edge p_0p_n cannot possibly cross any of them and, hence, is either above or below all the points in between. Therefore, a chain always admits a spanning cycle of unavoidable edges with at least one hull edge. We prove in Section 2 that this is also a characterization of chains in terms of *order types* (see [14] for a definition).

► **Theorem 2.** *For every point set that admits a spanning cycle of unavoidable edges including at least one convex hull edge, there exists a chain with the same order type.*

All of the mentioned families of point sets (convex position, double chain, and so on) are usually neither defined nor depicted in a way that makes it clear that they may be thought of as chains as in Definition 1. Still, the premise of Theorem 2 is easily verified for all of them except for the double circle, which may however be transformed into a chain by removing one of the inner points. Figure 1 shows realizations of some such point sets as chains.

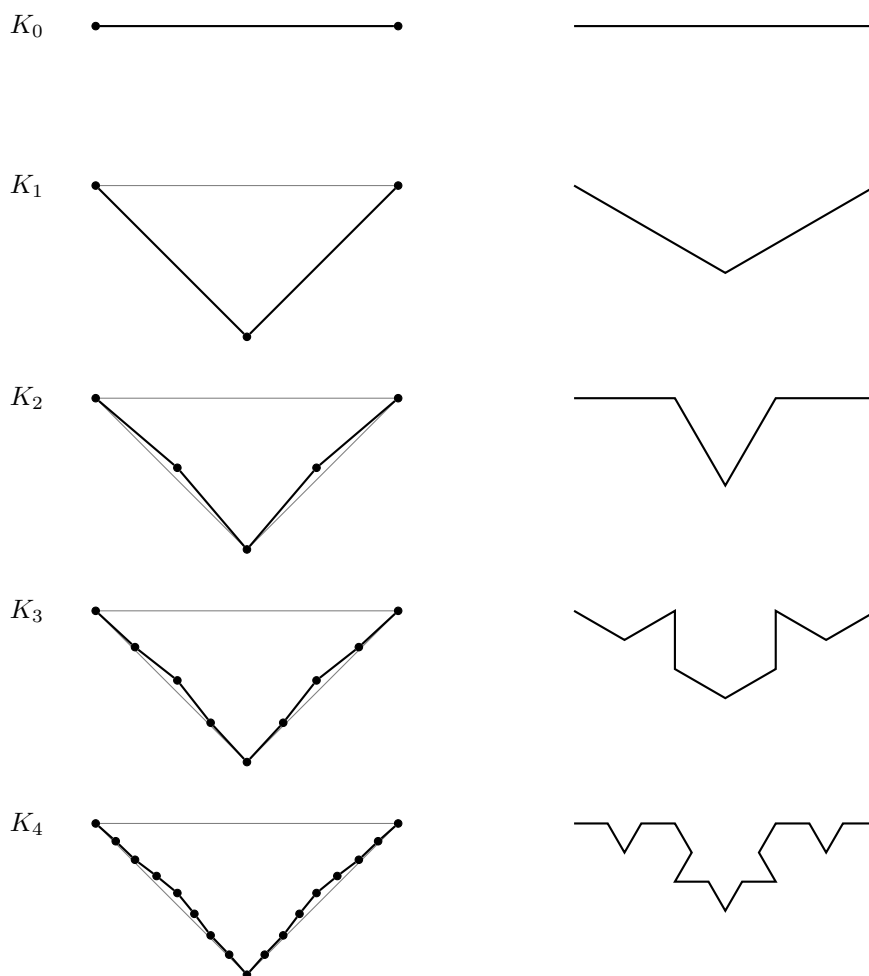
Imagine walking along the chain edges and recording at each point the information whether we make a left turn or a right turn. It can be noted already now that such information – while crucial – is not enough to really capture all of the relevant combinatorial structure of a given chain. Instead, the right way of looking at it turns out to be recording for each edge $p_i p_j$ whether it lies above or below all the chain edges in between.

The simple linear structure inherent to chains allows us to develop a combinatorial theory in Section 2, by which every chain admits a unique construction starting from the primitive chain with only one edge. Two types of sum operations, so-called convex and concave sums, are used to “concatenate” chains, while an inversion allows to “flip” a chain on its head. This yields for every chain a concise and unique description as an algebraic formula. Based on this, we will also see that the number of combinatorially different chains is equal to S_{n-1} , where $S_k = \sum_{i=0}^k \frac{1}{i+1} \binom{k}{i} \binom{k+i}{i} = \Theta(k^{-3/2}(3 + \sqrt{8})^k)$ is the k -th large Schröder number [2].

Triangulations of chains. The unavoidable chain edges separate every triangulation cleanly into an *upper triangulation* of the region above the chain edges and into a *lower triangulation* of the region below. Therefore, both upper and lower triangulations may be analyzed separately. It also follows that there is no further complication due to inner vertices as one would typically encounter them in general point sets.

There is a simple cubic time dynamic programming algorithm for counting triangulations of simple polygons [13]. Such an algorithm can of course also be used to count both the upper and lower triangulations of a given chain. However, we show in Section 3 that the additional structure of chains allows us to devise an improved quadratic time algorithm, which plays a crucial role in the derivation of our main result.

► **Theorem 3.** *Given a chain C with n chain edges as input, it is possible to compute the number $\text{tr}(C)$ by using only $O(n^2)$ integer additions and multiplications.*



■ **Figure 2** The Koch chains K_s for $s = 0, \dots, 4$ and the corresponding Koch curves. Even though it is hard to recognize for larger values of s , the changes in direction along the Koch curve on the right are reflected one-to-one by the chain edges of the corresponding Koch chain on the left.

The Koch chain. There is a particular type of chain that has caught our interest and which, to the best of our knowledge, has not been described in the literature before. We call it the *Koch chain* due to its striking similarity in appearance and definition to the famous Koch curve. More precise definitions follow later in Definition 14; for now, suppose K_0 is a primitive chain with just one chain edge, and let the s -th iteration K_s of the Koch chain be defined by concatenating two flipped and sufficiently flattened copies of K_{s-1} in such a way that the chain edges at the point of concatenation form a left turn, see Figure 2.

Koch chains turn out to have a particularly large number of triangulations, much more so than any other known point sets. For values of s up to 21, we have computed the corresponding numbers of upper and lower triangulations, as well as complete triangulations, by using our algorithm from Theorem 3. The results are displayed in Table 1.

In consequence, concatenating copies of K_{21} side by side results in an infinite family of point sets with at least 9.082798^n triangulations. This alone already establishes the improved lower bound of $\text{tr}_{\max}(n) = \Omega(9.082798^n)$.

■ **Table 1** The computed numbers of triangulations of the Koch chain K_s for $s = 0, \dots, 21$. As usual, n is the number of chain edges, whereas U , L , and T stand, respectively, for the numbers of upper, lower, and complete triangulations of the corresponding Koch chain.

s	n	$\sqrt[n]{U}$	$\sqrt[n]{L}$	$\sqrt[n]{T}$	s	n	$\sqrt[n]{U}$	$\sqrt[n]{L}$	$\sqrt[n]{T}$
0	1	1.0	1.0	1.0	11	2048	3.121029	2.858643	8.921910
1	2	1.0	1.0	1.0	12	4096	2.882177	3.121029	8.995359
2	4	1.189207	1.0	1.189207	13	8192	3.134955	2.882177	9.035496
3	8	1.791279	1.189207	2.130201	14	16384	2.889213	3.134955	9.057554
4	16	2.035453	1.791279	3.646065	15	32768	3.139056	2.889213	9.069406
5	32	2.558954	2.035453	5.208633	16	65536	2.891256	3.139056	9.075820
6	64	2.564646	2.558954	6.562814	17	131072	3.140236	2.891256	9.079229
7	128	2.935733	2.564646	7.529118	18	262144	2.891838	3.140236	9.081055
8	256	2.783587	2.935733	8.171870	19	524288	3.140569	2.891838	9.082019
9	512	3.075469	2.783587	8.560839	20	1048576	2.892001	3.140569	9.082530
10	1024	2.858643	3.075469	8.791671	21	2097152	3.140662	2.892001	9.082799

Poly chains and Twin chains. We were unable to nail down the exact asymptotic behavior of the number of triangulations of K_s as s approaches infinity. It is also unclear how much is lost due to undercounting by not considering any interactions between the different copies of K_{21} in our simple lower bound construction from just before.

To remedy the situation somewhat, in Section 4 we define and analyze more carefully the *poly-C chain* (a specific way of concatenating k copies of a given chain C) and the *twin-C chain* (a construction where two copies of a poly- C chain face each other, similar in spirit to the classic double chain). Based on these considerations, we get a slightly improved lower bound construction, and we are also able to conclude that the numbers in the last column of Table 1 will not grow significantly larger than what we already have.

► **Theorem 4.** Let C_k be the twin- K_{21} chain that uses $2k$ copies of K_{21} in total. Then,

$$\lim_{k \rightarrow \infty} \sqrt[n]{\text{tr}(C_k)} = 9.083095\dots, \quad \text{tr}_{\max}(n) = \Omega(9.083095^n).$$

► **Theorem 5.** For the Koch chain K_s with $n = 2^s$ chain edges, we have

$$9.082798 \leq \lim_{s \rightarrow \infty} \sqrt[n]{\text{tr}(K_s)} \leq 9.083139.$$

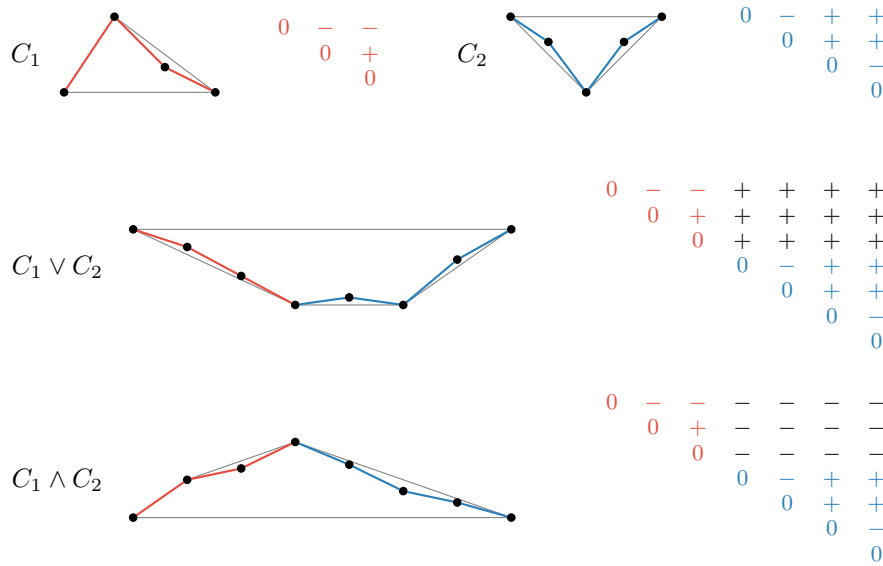
2 Structural Properties of Chains

Recall Definition 1 from the introduction. Note that the unavoidable chain edges form an x -monotone curve $p_0p_1 \dots p_n$, to which we refer as the *chain curve*. An edge $p_i p_j$ that is not a chain edge cannot cross the chain curve, and so it lies either above or below that curve.

► **Definition 6.** To every chain C we associate a visibility triangle $V(C)$ with entries

$$V(C)_{i,j} = \begin{cases} +1, & \text{if } p_i p_j \text{ lies above the chain curve;} \\ -1, & \text{if } p_i p_j \text{ lies below the chain curve;} \\ 0, & \text{if } p_i p_j \text{ is a chain edge (i.e., } i + 1 = j); \end{cases} \quad (0 \leq i < j \leq n).$$

As an example, the visibility triangles of the chains that correspond to the classic point sets from the introduction can be seen in Figure 3.



■ **Figure 5** In the top row, two chains C_1 and C_2 with their visibility triangles. Below, the corresponding convex and concave sums $C_1 \vee C_2$ and $C_1 \wedge C_2$. Red and blue color is used to highlight the contained substructures and their origin.

2.2 Convex and Concave Sums

Given two chains C_1 and C_2 , we would like to concatenate them so that we get a new chain containing C_1 and C_2 as substructures. As shown in Figure 5, there are two ways to do so.

► **Proposition 8.** *Let C_1 and C_2 be chains with n_1 and n_2 chain edges, respectively. Then, there is an upward chain (which we denote by $C_1 \vee C_2$ and call the convex sum of C_1 and C_2) with $n_1 + n_2$ chain edges and visibility triangle*

$$V(C_1 \vee C_2)_{i,j} = \begin{cases} V(C_1)_{i,j}, & \text{if } i, j \in [0, n_1]; \\ V(C_2)_{i-n_1, j-n_1}, & \text{if } i, j \in [n_1, n_1 + n_2]; \\ +1, & \text{if } i < n_1 < j; \end{cases} \quad (0 \leq i < j \leq n_1 + n_2).$$

► **Proposition 9.** *Similarly, there is a downward chain (which we denote by $C_1 \wedge C_2$ and call the concave sum of C_1 and C_2) with $n_1 + n_2$ chain edges and visibility triangle*

$$V(C_1 \wedge C_2)_{i,j} = \begin{cases} V(C_1)_{i,j}, & \text{if } i, j \in [0, n_1]; \\ V(C_2)_{i-n_1, j-n_1}, & \text{if } i, j \in [n_1, n_1 + n_2]; \\ -1, & \text{if } i < n_1 < j; \end{cases} \quad (0 \leq i < j \leq n_1 + n_2).$$

Proof of Proposition 8. We focus on the convex sum; the proof for the concave sum is analogous. We have to show that there is a point set that forms a chain with the specified visibility triangle. Intuitively speaking, this is achieved by first flattening the two given chains and then arranging them in a V-shape.

To be more precise, we employ vertical shearings, which are maps $(x, y) \mapsto (x, y + \lambda x)$ in \mathbb{R}^2 for some $\lambda \in \mathbb{R}$. Vertical shearings preserve signed areas and x -coordinates. Hence, if a point set realizes a specific chain, then so does its image under any vertical shearing.

With the help of an appropriate vertical shearing, we may realize C_1 as a point set in the rectangle $[-1, 0] \times [-1, 1]$ in such a way that the first point is at $(-1, 0)$ and the last point is at $(0, 0)$. Then, given any $\varepsilon \geq 0$, we may rescale vertically to get a point set $Q_1(\varepsilon)$ in the rectangle $[-1, 0] \times [-\varepsilon, \varepsilon]$. Let now $R_1(\varepsilon)$ be the image of $Q_1(\varepsilon)$ under the vertical shearing with $\lambda = -1$. Then, the first point of $R_1(\varepsilon)$ lies at $(-1, 1)$, while the last point remains at $(0, 0)$. For $\varepsilon > 0$, since $Q_1(\varepsilon)$ is a realization of C_1 , so is $R_1(\varepsilon)$. On the other hand, for $\varepsilon = 0$, the points of $R_1(\varepsilon)$ all lie on the segment between $(-1, 1)$ and $(0, 0)$.

With C_2 we proceed similarly to get a point set $Q_2(\varepsilon)$ in the rectangle $[0, 1] \times [-\varepsilon, \varepsilon]$, but we now apply the vertical shearing with $\lambda = 1$ to get $R_2(\varepsilon)$ with the first point at $(0, 0)$ and the last point at $(1, 1)$.

Let $T(\varepsilon) = R_1(\varepsilon) \cup R_2(\varepsilon)$. We claim that for $\varepsilon > 0$ small enough, $T(\varepsilon)$ is a chain with visibility triangle $V(C_1 \vee C_2)$ as specified. Indeed, as $R_i(\varepsilon)$ is a realization of C_i , we only need to check that the edges between any point of $R_1(\varepsilon)$ and any point of $R_2(\varepsilon)$ (excluding the common point at the origin) lie above all the points in between. Since this is the case for $\varepsilon = 0$ and $T(\varepsilon)$ depends continuously on ε , the claim follows. ◀

2.3 Algebraic Properties

Using the formulas for the visibility triangles from the corresponding transformations in Propositions 7–9, it can be checked easily that the following algebraic laws hold.

► **Lemma 10.** *Let C_1, C_2, C_3 be arbitrary chains. Then, the following are all true.*

$$\begin{aligned} \text{Involution:} & \quad \overline{\overline{C_1}} = C_1 \\ \text{De Morgan:} & \quad \overline{C_1 \vee C_2} = \overline{C_1} \wedge \overline{C_2}, \quad \overline{C_1 \wedge C_2} = \overline{C_1} \vee \overline{C_2} \\ \text{Associativity:} & \quad (C_1 \vee C_2) \vee C_3 = C_1 \vee (C_2 \vee C_3), \quad (C_1 \wedge C_2) \wedge C_3 = C_1 \wedge (C_2 \wedge C_3) \end{aligned}$$

However, note that for example $(C_1 \wedge C_2) \vee C_3$ is not the same chain as $C_1 \wedge (C_2 \vee C_3)$.

2.4 Examples

We denote by E the primitive chain with only $n = 1$ chain edge; that is, the visibility triangle has just the entry $V(E)_{0,1} = 0$. Using this as a building block, we may define two more fundamental chains, the *convex chain* $C_{\text{cvx}}(n)$ and the *concave chain* $C_{\text{ccv}}(n)$, by setting

$$C_{\text{cvx}}(n) = \underbrace{E \vee \cdots \vee E}_{n \text{ copies}}, \quad C_{\text{ccv}}(n) = \underbrace{E \wedge \cdots \wedge E}_{n \text{ copies}}.$$

The convex chain is an upward chain, while the concave chain is a downward chain. Also, since $\overline{E} = E$, we get $\overline{C_{\text{cvx}}(n)} = C_{\text{ccv}}(n)$ by using De Morgan’s law. Finally, note that $C_{\text{cvx}}(n)$ and $C_{\text{ccv}}(n)$ are distinct as chains, even though they both are in convex position.

As already mentioned in the introduction, many previously studied point sets are in fact chains, or can be seen as such. Using flips as well as convex and concave sums, we can now describe these configurations with very concise formulas.

► **Example 11.** The *double chain* with $n = 2k + 1$ chain edges is the chain

$$C_{\text{dbl}}(n) = C_{\text{ccv}}(k) \vee E \vee C_{\text{cvx}}(k).$$

► **Example 12.** The *zig-zag chain* with $n = 2k$ chain edges (which, in essence, is a double circle with one of the inner points removed) is the chain

$$C_{\text{zz}}(n) = \underbrace{C_{\text{ccv}}(2) \vee \cdots \vee C_{\text{ccv}}(2)}_{k \text{ copies}}.$$

► **Example 13.** The double zig-zag chain with $n = 4k + 1$ chain edges is the chain

$$C_{\text{dzz}}(n) = \overline{C_{\text{zz}}(2k)} \vee E \vee \overline{C_{\text{zz}}(2k)}.$$

All these examples involve formulas of constant nesting depth only. But the tools developed up to this point allow us to also define more complicated chains via formulas of non-constant nesting depth, without having to worry about questions of realizability. One such chain with logarithmic nesting depth is indeed the Koch chain.

► **Definition 14.** The Koch chain K_s is an upward chain with $n = 2^s$ chain edges, defined recursively via $K_0 = E$ and $K_s = \overline{K_{s-1}} \vee \overline{K_{s-1}}$ for all $s \geq 1$.

Indeed, after expanding the recursive definition twice and using De Morgan's law on both sides, we see that the formula $K_s = (K_{s-2} \wedge K_{s-2}) \vee (K_{s-2} \wedge K_{s-2})$ has a complete binary parse tree with alternating convex and concave sums on any path from the root to a leaf.

2.5 Unique Construction

We want to prove the following result. In essence, it states that every chain can be constructed in a unique way by using only convex and concave sums.

► **Theorem 15.** Every chain can be expressed as a formula involving convex sums, concave sums, parentheses, and copies of the primitive chain with only one chain edge. This formula is unique up to redundant parentheses (redundant due to associativity as in Lemma 10).

In particular, the above theorem allows us to encode a chain with $O(n)$ bits (as opposed to the $O(n^2)$ bits required for the visibility triangle) and to easily enumerate all chains of a fixed size. We further see that the number of upward chains is given by the little Schröder numbers [3] and the number of all chains is given by the large Schröder numbers [2].

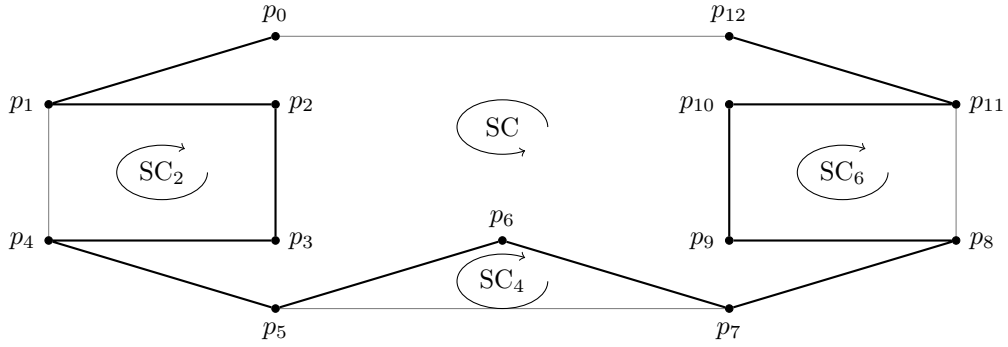
The theorem follows by induction from the following proposition (and from an analogous proposition that expresses downward chains as a unique concave sum of upward chains).

► **Proposition 16.** Let C be an upward chain with $n > 1$ chain edges. Suppose that the lower convex hull of C is $p_{i_0}p_{i_1}\dots p_{i_k}$ with $0 = i_0 < \dots < i_k = n$. For $j = 1, \dots, k$, let C_j be the chain with points $p_{i_{j-1}}, \dots, p_{i_j}$. Then, each C_j is a downward chain. Moreover, $C = C_1 \vee \dots \vee C_k$ and any formula that evaluates to C has the same top-level structure.

Proof. As $p_{i_{j-1}}p_{i_j}$ is an edge of the lower convex hull of C , it is below all the points in between. Hence, each C_j is indeed a downward chain.

To prove $C = C_1 \vee \dots \vee C_k$, we have to show that both chains have the same visibility triangle. By definition of the C_j , the visibility triangles clearly agree on all entries that stem from an edge $p_a p_b$ where p_a and p_b are both part of the same C_j . On the other hand, if p_a and p_b are not part of the same C_j , then there is a j with $a < i_j < b$. As p_{i_j} is a vertex of the lower convex hull, it lies below the edge $p_a p_b$ and hence $V(C)_{a,b} = +1$. But this is precisely what we also get for the visibility triangle of the convex sum $C_1 \vee \dots \vee C_k$.

For uniqueness, suppose we are given any formula for C . Since C is assumed to be an upward chain and since any concave sum is a downward chain, the formula must be of the form $C'_1 \vee \dots \vee C'_{k'}$. We may further assume that each C'_j is a downward chain by omitting redundant parentheses. Observe now that in any such convex sum of downward chains, the resulting lower convex hull is determined by the points that are shared by any two consecutive chains C'_j . Since the given formula evaluates to C , we must have $k' = k$ and $C'_j = C_j$. ◀



■ **Figure 6** The situation in the proof of Theorem 2. Beware that this is just a sketch; in reality, the pockets would need to be much more narrow in order to make all edges of SC unavoidable.

2.6 Geometric Characterization

As already mentioned in the introduction, the chain edges together with the hull edge p_0p_n form a spanning cycle of unavoidable edges. We are now ready to prove that this property characterizes chains geometrically.

Proof of Theorem 2. Let $SC = p_0p_1 \dots p_n$ be the spanning cycle in counter-clockwise order, with p_0p_n an edge of the convex hull, which we call the *base edge*. As SC consists of unavoidable edges only, it cannot be crossed by any edge that is not part of SC . Hence, we can associate a visibility triangle with the given point set, similar to the visibility triangle of a chain, by setting

$$V_{i,j} = \begin{cases} +1, & \text{if } p_i p_j \text{ is inside } SC \text{ or the base edge;} \\ -1, & \text{if } p_i p_j \text{ is outside } SC; \\ 0, & \text{if } p_i p_j \text{ is part of } SC \text{ (i.e., } i+1=j); \end{cases} \quad (0 \leq i < j \leq n).$$

By using that p_0p_n is a hull edge and by some geometric considerations², one can then show that for $i < j < k$, the triangle $p_i p_j p_k$ is oriented counter-clockwise if and only if $V_{i,k} = +1$. Hence, it suffices to construct a chain whose visibility triangle agrees with V .

Let $p_{i_0}, p_{i_1}, \dots, p_{i_k}$ be the vertices of the convex hull with $0 = i_0 < \dots < i_k = n$. For $1 \leq j \leq k$, let $P_j = \{p_{i_{j-1}}, \dots, p_{i_j}\}$. We now see that either P_j consists of only two points or that it admits a spanning cycle of unavoidable edges, namely $SC_j = p_{i_{j-1}}p_{i_{j-1}+1} \dots p_{i_j}$ with base edge $p_{i_{j-1}}p_{i_j}$. The situation is depicted in Figure 6. Note that the inside of SC_j is outside of SC . In fact, SC_j forms a so-called pocket of SC , which means that all edges of the cycle SC_j except for $p_{i_{j-1}}p_{i_j}$ are also edges of SC .

By induction, there is a chain C_j with the same order type as P_j , that is, with

$$V(C_j)_{a,b} = \begin{cases} +1, & \text{if } p_{i_{j-1}+a}p_{i_{j-1}+b} \text{ is inside } SC_j \text{ or the base edge;} \\ -1, & \text{if } p_{i_{j-1}+a}p_{i_{j-1}+b} \text{ is outside } SC_j; \\ 0, & \text{if } p_{i_{j-1}+a}p_{i_{j-1}+b} \text{ is part of } SC_j \text{ (i.e., } a+1=b); \end{cases} \quad (0 \leq a < b \leq i_j - i_{j-1}).$$

² This involves a lengthy case distinction that does not add much insight. We omit the details here.

Let us consider the flipped version \overline{C}_j . As noted before, the inside of SC_j is outside of SC. As SC_j moreover forms a pocket of SC, any edge outside of SC_j is inside SC. Hence,

$$V(\overline{C}_j)_{a,b} = \begin{cases} +1, & \text{if } p_{i_{j-1}+a}p_{i_{j-1}+b} \text{ is inside SC;} \\ -1, & \text{if } p_{i_{j-1}+a}p_{i_{j-1}+b} \text{ is outside SC;} \\ 0, & \text{if } p_{i_{j-1}+a}p_{i_{j-1}+b} \text{ is part of SC (i.e., } a + 1 = b); \end{cases} \quad (0 \leq a < b \leq i_j - i_{j-1}).$$

We claim that $C = \overline{C}_1 \vee \dots \vee \overline{C}_k$ has the desired visibility triangle V . We have just seen that the entries stemming from the individual \overline{C}_j are correct. So, all that is left to observe is that edges between different pockets lie inside of SC, which is indeed the case. ◀

3 Triangulations of Chains

In the previous section, we have seen that any chain can be expressed as a formula involving only convex and concave sums. Our goal here is to understand how triangulations behave with respect to such convex and concave sums. In order for this to work out, we have to consider not just triangulations, but a more general notion of partial triangulations.

We start by decomposing triangulations of a chain C into an upper and a lower part. An edge $p_i p_j$ is an *upper edge* if $V(C)_{i,j} = +1$, a *chain edge* if $V(C)_{i,j} = 0$, and a *lower edge* if $V(C)_{i,j} = -1$. That is, upper edges lie above the chain curve, while lower edges lie below.

► **Definition 17.** An upper (lower) triangulation of a given chain C is a crossing-free geometric graph on C that is edge-maximal subject to only containing chain edges and upper (lower) edges. We denote the number of upper and lower triangulations by $U(C)$ and $L(C)$, respectively, and as always the number of (complete) triangulations by $\text{tr}(C)$.

Note that the chain edges are contained in every upper and lower triangulation. Moreover, every triangulation is the union of a unique upper and a unique lower triangulation, which implies $\text{tr}(C) = U(C) \cdot L(C)$. A lower triangulation of a chain C is an upper triangulation of the flipped version \overline{C} , and therefore $L(C) = U(\overline{C})$. For this reason, we may restrict our attention to studying only upper triangulations.

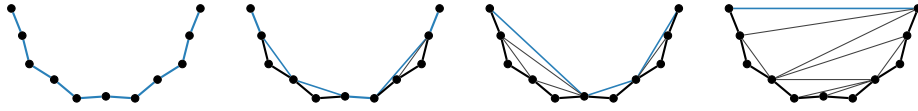
Intuitively speaking, we can create a partial upper triangulation by combining all the chain edges with some upper edges, in such a way that all bounded faces are triangles. Note that then, only some of the used edges are visible from above.

► **Definition 18.** Let C be any chain with n chain edges, and let $V = p_{i_0} p_{i_1} \dots p_{i_v}$ with $0 = i_0 < i_1 < \dots < i_v = n$ be an (x -monotone) curve composed of chain edges and upper edges only. A partial upper triangulation of C (with visible edges V) consists of all chain edges, all edges in V , and a triangulation of the areas between the two.

Figure 7 depicts some partial upper triangulations and their visible edges. We are interested in counting such triangulations parameterized by the number of triangles. It can be noted that a partial upper triangulation with k triangles has $n - k$ visible edges.

► **Definition 19.** Let C be any chain with n chain edges. For $k = 0, \dots, n - 1$, let $t_k(C)$ be the number of partial upper triangulations of C with k triangles (i.e., with $n - k$ visible edges). The upper triangulation polynomial of C is the corresponding generating function

$$T_C(x) = \sum_{k=0}^{n-1} t_k(C) x^k.$$



■ **Figure 7** Four partial upper triangulations of the “double circle” with ten, six, three, and one visible edge, respectively. As usual, chain edges are in bold, while visible edges are in blue.

As an example, enumerating all partial upper triangulations of the convex chain $C_{\text{cvx}}(4)$ shows that $T_{C_{\text{cvx}}(4)}(x) = 1 + 3x + 5x^2 + 5x^3$. In general, note that for every chain C we have $t_0(C) = 1$ and that the leading coefficient of $T_C(x)$ is equal to $U(C)$. Moreover, we may again think of $T_{\overline{C}}(x)$ as the “lower triangulation polynomial” of C .

3.1 Convex and Concave Sums

Let us start with the easy case. For concave sums, we can establish the following relation.

► **Lemma 20.** *A partial upper triangulation of $C_1 \wedge C_2$ is the union of a unique partial upper triangulation of C_1 and a unique partial upper triangulation of C_2 . Hence,*

$$T_{C_1 \wedge C_2}(x) = T_{C_1}(x) \cdot T_{C_2}(x), \quad U(C_1 \wedge C_2) = U(C_1) \cdot U(C_2).$$

Convex sums are more tricky. The main insight is that every partial upper triangulation of $C_1 \vee C_2$ consists of a partial upper triangulation of C_1 , a partial upper triangulation of C_2 , and some edges between C_1 and C_2 . More precisely:

- **Proposition 21.** *There is a triangle-preserving bijection between*
- *all triples (T_1, T_2, T_3) where T_1 is a partial upper triangulation of C_1 (with v_1 visible edges), T_2 is a partial upper triangulation of C_2 (with v_2 visible edges), and T_3 is a partial upper triangulation of the convex sum $C_{\text{ccv}}(v_1) \vee C_{\text{ccv}}(v_2)$, and*
 - *all partial upper triangulations of $C_1 \vee C_2$.*

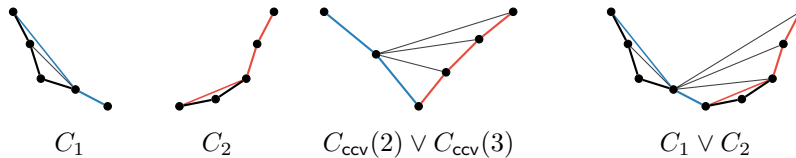
This bijection is defined by taking the union of all triangles, see Figure 8. The proposition then directly implies the following equation for the upper triangulation polynomial.

► **Lemma 22.** *Let C_1 and C_2 be chains with n_1 and n_2 chain edges, respectively. Then,*

$$T_{C_1 \vee C_2}(x) = \sum_{k_1=0}^{n_1-1} \sum_{k_2=0}^{n_2-1} t_{k_1}(C_1) \cdot t_{k_2}(C_2) \cdot x^{k_1+k_2} \cdot T_{C_{\text{ccv}}(n_1-k_1) \vee C_{\text{ccv}}(n_2-k_2)}(x).$$

Let us consider the special case of a convex sum of two concave chains with n_1 and n_2 chain edges, respectively. Note that any partial upper triangulation of such a chain has at most one upper edge that is visible. Summing over all possibilities for that edge, we get

$$T_{C_{\text{ccv}}(n_1) \vee C_{\text{ccv}}(n_2)}(x) = 1 + \sum_{l=1}^{n_1} \sum_{r=1}^{n_2} \binom{l+r-2}{l-1} x^{l+r-1}.$$



■ **Figure 8** From left to right, the respective partial upper triangulations T_1 of C_1 , T_2 of C_2 , T_3 of $C_{\text{ccv}}(2) \vee C_{\text{ccv}}(3)$, and the resulting partial upper triangulation of $C_1 \vee C_2$ as in Proposition 21.

Combining the above equation with Lemma 22 allows us to compute $T_{C_1 \vee C_2}(x)$ from $T_{C_1}(x)$ and $T_{C_2}(x)$. Furthermore, by comparing the leading coefficients in the formulas from Lemmas 20 and 22, we get the following obvious but important fact.

► **Corollary 23.** $C_1 \vee C_2$ has at least as many upper triangulations as $C_1 \wedge C_2$. That is,

$$U(C_1 \vee C_2) \geq U(C_1 \wedge C_2).$$

Finally, note that the two chains $C_1 \vee C_2$ and $C_2 \vee C_1$ can be quite different from a geometric point of view. But in terms of the number of triangulations, they are the same.

► **Corollary 24.** For any two chains C_1 and C_2 , we have

$$T_{C_1 \vee C_2}(x) = T_{C_2 \vee C_1}(x), \quad T_{C_1 \wedge C_2}(x) = T_{C_2 \wedge C_1}(x).$$

3.2 Dynamic Programming

In this subsection, we show how to use dynamic programming in order to speed up the computations for a convex sum. To simplify the analysis, we assume a computational model where all additions and multiplications take only constant time.

► **Proposition 25.** Let C_1 and C_2 be chains with n_1 and n_2 chain edges, respectively. Given the coefficients of $T_{C_1}(x)$ and $T_{C_2}(x)$, we can compute $T_{C_1 \vee C_2}(x)$ in $O(n_1 n_2)$ time.

Recall that by Theorem 15, we can write any chain C as a formula involving only convex sums, concave sums, and primitive chains with only one chain edge. Therefore, using Proposition 25 for convex sums and Lemma 20 for concave sums, we are able to compute $T_C(x)$ in quadratic time. Clearly, this proves Theorem 3 from the introduction.

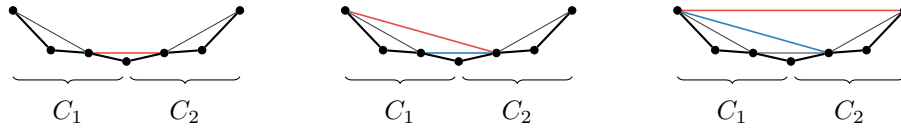
Proof of Proposition 25. Observe that every partial upper triangulation of $C_1 \vee C_2$ either corresponds to a partial upper triangulation of $C_1 \wedge C_2$, or it has a unique visible upper edge that connects a vertex of C_1 with a vertex of C_2 . Let us call this edge the *bridge*. Let further $\text{DP}[l][r]$ be the number of partial upper triangulations whose visible edges consist of l visible edges in C_1 , followed by the bridge, followed by r visible edges in C_2 . Then,

$$T_{C_1 \vee C_2}(x) = T_{C_1 \wedge C_2}(x) + \sum_{l=0}^{n_1-1} \sum_{r=0}^{n_2-1} \text{DP}[l][r] \cdot x^{n_1+n_2-l-r-1}.$$

To compute the table DP, let us see what happens when we remove the bridge. We either end up with a partial upper triangulation of $C_1 \wedge C_2$ with $l + 1$ and $r + 1$ visible edges in C_1 and C_2 , respectively, or we get a new bridge, which used to be an edge of the triangle below the old bridge. In the latter case, depending on which of the two possible edges this is, we end up with one more visible edge in either C_1 or C_2 . Figure 9 depicts these three cases. To summarize, for all l and r ($0 \leq l < n_1, 0 \leq r < n_2$),

$$\text{DP}[l][r] = t_{n_1-l-1}(C_1) \cdot t_{n_2-r-1}(C_2) + \text{DP}[l+1][r] + \text{DP}[l][r+1],$$

with the base case $\text{DP}[n_1][r] = \text{DP}[l][n_2] = 0$. Therefore, filling up the table DP takes $O(n_1 n_2)$ time, as desired. ◀



■ **Figure 9** The three cases when removing the bridge from a partial upper triangulation of $C_1 \vee C_2$ in the proof of Proposition 25. On the left, both C_1 and C_2 gain a visible edge. In the middle, only C_1 gains a visible edge. On the right, only C_2 gains a visible edge. The current bridge is red, and the edge that becomes the new bridge is blue.

3.3 Koch Chains

Recall Definition 14 and that the formula for Koch chains expands to the nested expression

$$K_s = (K_{s-2} \wedge K_{s-2}) \vee (K_{s-2} \wedge K_{s-2})$$

with alternating convex and concave sums. This repeated mixing of the two types of sums appears to make an exact analysis of the number of triangulations of K_s very difficult.

Instead, we have implemented the quadratic time algorithm from the previous subsection and used it to compute $T_{K_s}(x)$ and $T_{\overline{K_s}}(x)$ for all $s \leq 21$. To deal with the exponentially growing coefficients, we rely on a custom floating point type with a 64 bit mantissa and a 32 bit exponent from the boost multiprecision library. As only additions and multiplications are involved, we do not have to deal with numerical issues; in fact, the rounding errors grow at most linearly. In addition, we make use of multi-threading and take advantage of symmetries of K_s for a constant factor speed-up. This allows us to compute $T_{K_{21}}(x)$ in around a day on a regular workstation (Intel i7-6700HQ, 2.6GHz).

Table 1 from the introduction lists the resulting numbers. For example, K_{21} has approximately 9.082799^n triangulations, where $n = 2^{21}$. In the next section, we show how the computed coefficients of $T_{K_{21}}(x)$ can be used to give bounds on $\text{tr}(K_s)$ as $s \rightarrow \infty$.

4 Poly Chains and Twin Chains

Let C_0 be a chain with m chain edges. We want to define two particular families of chains that can be built from many copies of C_0 via concave and convex sums.

► **Definition 26.** For $N \geq 1$, the poly- C_0 chains (of length $n = mN$) are the chains

$$C_{\text{poly}}(C_0, N) = \underbrace{\overline{C_0} \vee \cdots \vee \overline{C_0}}_{N \text{ copies}}$$

► **Definition 27.** For $N \geq 1$, the twin- C_0 chains (of length $n = 2mN + 1$) are the chains

$$C_{\text{twin}}(C_0, N) = \overline{C_{\text{poly}}(C_0, N)} \vee E \vee \overline{C_{\text{poly}}(C_0, N)}.$$

Note that both resulting chains are upward chains, as long as $N > 1$. For example, the poly- E chains are the convex chains, the twin- E chains are the classic double chains, and the twin- $(E \vee E)$ chains are the double zig-zag chains.

We are interested in the asymptotic behavior of the number of triangulations of these constructions as $N \rightarrow \infty$. Lemma 20 gives us the number of lower triangulations.

$$\begin{aligned} L(C_{\text{poly}}(C_0, N)) &= U(C_0 \wedge \cdots \wedge C_0) = U(C_0)^N \\ L(C_{\text{twin}}(C_0, N)) &= U(C_{\text{poly}}(C_0, N) \wedge E \wedge C_{\text{poly}}(C_0, N)) = U(C_{\text{poly}}(C_0, N))^2 \end{aligned}$$

For the upper triangulations, we make use of the following general result.

► **Theorem 28.** *The chains $C_{\text{poly}}(C_0, N)$ have $\tilde{\Theta}(\lambda^n)$ upper triangulations, while the chains $C_{\text{twin}}(C_0, N)$ have $\tilde{\Theta}(\tau^n)$ upper triangulations, where*

$$\lambda = \sqrt[m]{\sum_{k=1}^m 2^k(k+1) \cdot t_{m-k}(\overline{C_0})}, \quad \tau = \sqrt[m]{\sum_{k=1}^m 2^k \cdot t_{m-k}(C_0)}.$$

It follows that the chains $C_{\text{twin}}(C_0, N)$ have $\tilde{\Theta}((\lambda\tau)^n)$ complete triangulations.

► **Example 29.** Let us analyze the poly- $C_{\text{cvx}}(4)$ chains and twin- $C_{\text{cvx}}(4)$ chains. We have

$$T_{C_{\text{cvx}}(4)}(x) = 1, \quad T_{C_{\text{twin}}(4)}(x) = 1 + 3x + 5x^2 + 5x^3,$$

which yields $\lambda = \sqrt[4]{80}$ and $\tau = \sqrt[4]{70}$. Therefore, the twin- $C_{\text{cvx}}(4)$ chains have $\tilde{\Theta}(\sqrt[4]{5600}^n)$ triangulations, where $\sqrt[4]{5600} \approx 8.6506154$. Note that these chains are the generalized double zig-zag chains from [12]. By comparison, the numerical bound there was $\tilde{\Omega}(8.6504^n)$.

Using the coefficients of $T_{K_{21}}(x)$ and $T_{\overline{K_{21}}}(x)$ that we computed with our algorithm, we can also analyze the twin- K_{21} chains and, therefore, prove Theorem 4 from the introduction.

► **Corollary 30.** *The chains $C_{\text{twin}}(K_{21}, N)$ have $\tilde{\Theta}(\lambda^n)$ triangulations, for $\lambda \approx 9.083095$.*

The next lemma, combined with the first part of Theorem 28, can further be used to show asymptotic upper bounds for families of chains that are built from the same C_0 .

► **Lemma 31.** *Let C be any chain that can be written as a formula involving convex sums, concave sums and exactly N copies of C_0 . Then,*

$$U(C_0)^N \leq U(C) \leq U(C_{\text{poly}}(\overline{C_0}, N)).$$

Proof. Use induction on N with Corollary 23. ◀

► **Corollary 32.** *In the same setting, we have*

$$\text{tr}(C_0)^N \leq \text{tr}(C) \leq U(C_{\text{poly}}(C_0, N)) \cdot U(C_{\text{poly}}(\overline{C_0}, N)).$$

Proof. Apply Lemma 31 twice. First to C with C_0 , then to \overline{C} with $\overline{C_0}$. ◀

The Koch chains K_s with $s \geq 21$ can be written as formulas involving copies of K_{21} , so Corollary 32 applies to them. We get $9.082798^n \leq \text{tr}(K_s) \leq 9.083139^n$, as in Theorem 5.

4.1 Tools for the proof of Theorem 28

We only sketch the main steps here. We use similar ideas as Section 2 of [12] with three improvements that yield an exact $\tilde{\Theta}$ instead of a numerical lower bound. The first improvement is that our chain framework allows us to analyze even more general “double circles”.

► **Theorem 33.** *Let $c_1, \dots, c_m \geq 0$ be integers. Define*

$$V(c_1, \dots, c_m) = C_{\text{poly}}(C_{\text{cvx}}(1), c_1) \vee \dots \vee C_{\text{poly}}(C_{\text{cvx}}(m), c_m)$$

where we omit poly chains with $c_k = 0$. Then,

$$U(V(c_1, \dots, c_m)) \in \tilde{\Omega}\left(\prod_{k=1}^m (2^k(k+1))^{c_k}\right)$$

where the polynomial factors in the $\tilde{\Omega}$ only depend on m (and not on the c_k).

59:16 Chains, Koch Chains, and Point Sets with Many Triangulations

Proof. By Corollary 23 and Corollary 24, we get

$$U(V(c_1, \dots, c_m)) \geq U(C_{\text{poly}}(C_{\text{cvx}}(1), c_1)) \cdots U(C_{\text{poly}}(C_{\text{cvx}}(m), c_m)).$$

In [8] it is shown that $U(C_{\text{poly}}(C_{\text{cvx}}(k), N)) \in \tilde{\Omega}((2^k(k+1))^N)$. \blacktriangleleft

The second improvement is to replace the numerical optimization in [12] by this lemma.

► **Lemma 34.** *Let $u_1, \dots, u_m \geq 0$ be given. Let $H(\alpha_1, \dots, \alpha_m) = -\sum_k \alpha_k \ln \alpha_k$ be the entropy function. Then,*

$$\max_{\substack{0 \leq \alpha_1, \dots, \alpha_m \leq 1 \\ \alpha_1 + \dots + \alpha_m = 1}} e^{H(\alpha_1, \dots, \alpha_m)} \cdot \prod_{k=1}^m u_k^{\alpha_k} = \sum_{k=1}^m u_k.$$

Proof. Without loss of generality, assume that $u_k > 0$. Then, by Lagrange multipliers, the only maximum is at $\alpha_k = u_k / (u_1 + \dots + u_m)$. \blacktriangleleft

The third improvement is a special type of generating function that behaves well with regards to convex sums, allowing us to prove a matching upper bound for Theorem 33.

► **Definition 35.** *Let C be a chain of length n . The triangulation generating function is*

$$\phi_C(x) := T_C(x) - \left(\frac{x}{1-x}\right)^{n+1} T_C(1-x).$$

Note that $\phi_C(x)$ is a rational function. As a formal power series, $\phi_C(x) = T_C(x) + O(x^{n+1})$.

► **Theorem 36.** *For any two chains C_1 and C_2 , we have*

$$\phi_{C_1 \vee C_2}(x) = \phi_{C_1}(x) \cdot \phi_{C_2}(x) \cdot \frac{1-x}{1-2x}.$$

Proof. By Lemma 22, it suffices to prove this for $C_i = C_{\text{ccv}}(n_i)$. We have

$$\phi_{C_{\text{ccv}}(n)}(x) = 1 - \left(\frac{x}{1-x}\right)^{n+1}, \quad T_{C_{\text{ccv}}(n_1) \vee C_{\text{ccv}}(n_2)}(x) = 1 + \sum_{l=1}^{n_1} \sum_{r=1}^{n_2} \binom{l+r-2}{l-1} x^{l+r-1}.$$

Then, induction on (n_1, n_2) and raw computations on power series suffice. \blacktriangleleft

► **Corollary 37.** *We have*

$$U(V(c_1, \dots, c_m)) \leq \prod_{k=1}^m (2^k(k+1))^{c_k}.$$

Proof. Let $n = c_1 + 2c_2 + \dots + mc_m$ be the length of $V(c_1, \dots, c_m)$. Theorem 36 allows us to compute $\phi_{V(c_1, \dots, c_m)}$. We have $U(V(c_1, \dots, c_m)) = [x^{n-1}] \phi_{V(c_1, \dots, c_m)}(x)$, so we compute

$$\begin{aligned} [x^{n-1}] \phi_{V(c_1, \dots, c_m)}(x) &= [x^{n-1}] \left(\frac{1-x}{1-2x}\right)^{c_1 + \dots + c_m - 1} \cdot \prod_{k=1}^m \left(\phi_{C_{\text{ccv}}(k)}(x)\right)^{c_k} \\ &= [x^{n-1}] \frac{1-2x}{1-x} \prod_{k=1}^m \left(\sum_{i=0}^k \left(\frac{x}{1-x}\right)^i\right)^{c_k} \leq [x^{n-1}] \prod_{k=1}^m \left(\sum_{i=0}^k \left(\frac{x}{1-x}\right)^i\right)^{c_k} \leq 2^n \prod_{k=1}^m (k+1)^{c_k} \end{aligned}$$

as expanding the second to last term gives us $\prod (k+1)^{c_k}$ summands, each some power of $\frac{x}{1-x}$, the x^{n-1} -coefficient of which is always less than 2^n . \blacktriangleleft

4.2 Proof of Theorem 28 (only Poly Chains)

Using Lemma 22, we can expand $T_{C_{\text{poly}}(C_0, N)}(x)$ into an N -fold sum where each summand is a product of N triangulation numbers t_{k_i} and some $T_{V(a_1, \dots, a_m)}(x)$. After grouping together summands with the same monomial of triangulation numbers, the leading coefficients are

$$U(C_{\text{poly}}(C_0, N)) = \sum_{\substack{0 \leq a_1, \dots, a_m \leq N \\ a_1 + \dots + a_m = N}} \binom{N}{a_1, \dots, a_m} \prod_{k=1}^m t_{m-k}(C_0)^{a_k} \cdot U(V(a_1, \dots, a_m)).$$

Then, on one hand, by Corollary 37 and the multinomial theorem,

$$\begin{aligned} U(C_{\text{poly}}(C_0, N)) &\leq \sum_{\substack{0 \leq a_1, \dots, a_m \leq N \\ a_1 + \dots + a_m = N}} \binom{N}{a_1, \dots, a_m} \prod_{k=1}^m t_{m-k}(C_0)^{a_k} \cdot \prod_{k=1}^m \left(2^k(k+1)\right)^{a_k} \\ &\leq \left(\sum_{k=1}^m 2^k(k+1) \cdot t_{m-k}(C_0)\right)^N. \end{aligned}$$

On the other hand, by Theorem 33 and the entropy bound for multinomial coefficients,

$$U(C_{\text{poly}}(C_0, N)) \geq \frac{1}{N^{c(m)}} \sum_{\substack{0 \leq a_1, \dots, a_m \leq N \\ a_1 + \dots + a_m = N}} e^{H\left(\frac{a_1}{N}, \dots, \frac{a_m}{N}\right)} \prod_{k=1}^m t_{m-k}(C_0)^{a_k} \cdot \prod_{k=1}^m \left(2^k(k+1)\right)^{a_k}.$$

By picking the largest summand, given by Lemma 34, we get the lower bound



$$U(C_{\text{poly}}(C_0, N)) \in \tilde{\Omega}\left(\left(\sum_{k=1}^m t_{m-k}(C) \cdot 2^k(k+1)\right)^N\right).$$

References

- 1 OEIS Foundation Inc. (2021). The on-line encyclopedia of integer sequences. Catalan numbers. URL: <https://oeis.org/A000108>.
- 2 OEIS Foundation Inc. (2021). The on-line encyclopedia of integer sequences. Large Schröder numbers. URL: <https://oeis.org/A006318>.
- 3 OEIS Foundation Inc. (2021). The on-line encyclopedia of integer sequences. Little Schröder numbers. URL: <https://oeis.org/A001003>.
- 4 Oswin Aichholzer, Victor Alvarez, Thomas Hackl, Alexander Pilz, Bettina Speckmann, and Birgit Vogtenhuber. An improved lower bound on the minimum number of triangulations. In *Proceedings of the 32nd International Symposium on Computational Geometry*, 2016. doi:10.4230/LIPIcs.SoCG.2016.7.
- 5 Oswin Aichholzer, Thomas Hackl, Clemens Huemer, Ferran Hurtado, Hannes Krasser, and Birgit Vogtenhuber. On the number of plane geometric graphs. *Graphs Comb.*, 23(Supplement-1):67–84, 2007. doi:10.1007/s00373-007-0704-5.
- 6 Miklós Ajtai, Václav Chvátal, Monroe M. Newborn, and Endre Szemerédi. Crossing-free subgraphs. In *Theory and Practice of Combinatorics*, volume 60 of *North-Holland Mathematics Studies*, pages 9–12. North-Holland, 1982.
- 7 Victor Alvarez and Raimund Seidel. A simple aggregative algorithm for counting triangulations of planar point sets and related problems. In *Proceedings of the 29th the Symposium on Computational Geometry*, 2013. doi:10.1145/2462356.2462392.
- 8 Andrei Asinowski, Christian Krattenthaler, and Toufik Mansour. Counting triangulations of some classes of subdivided convex polygons. *Eur. J. Comb.*, 62:92–114, 2017. doi:10.1016/j.ejc.2016.12.002.

- 9 David Avis and Komei Fukuda. Reverse search for enumeration. *Discret. Appl. Math.*, 65(1-3):21–46, 1996. doi:10.1016/0166-218X(95)00026-N.
- 10 Sergei Bespamyatnikh. An efficient algorithm for enumeration of triangulations. *Comput. Geom.*, 23(3):271–279, 2002. doi:10.1016/S0925-7721(02)00111-6.
- 11 Markus Denny and Christian Sohler. Encoding a triangulation as a permutation of its point set. In *Proceedings of the 9th Canadian Conference on Computational Geometry*, 1997.
- 12 Adrian Dumitrescu, André Schulz, Adam Sheffer, and Csaba D. Tóth. Bounds on the maximum multiplicity of some common geometric graphs. *SIAM J. Discret. Math.*, 27(2):802–826, 2013. doi:10.1137/110849407.
- 13 Peter Epstein and Jörg-Rüdiger Sack. Generating triangulations at random. *ACM Trans. Model. Comput. Simul.*, 4(3):267–278, 1994. doi:10.1145/189443.189446.
- 14 Jacob E. Goodman and Richard Pollack. Multidimensional sorting. *SIAM J. Comput.*, 12(3):484–507, 1983. doi:10.1137/0212032.
- 15 Ferran Hurtado and Marc Noy. Counting triangulations of almost-convex polygons. *Ars Comb.*, 45, 1997.
- 16 Dániel Marx and Tillmann Miltzow. Peeling and nibbling the cactus: Subexponential-time algorithms for counting triangulations and related problems. In *Proceedings of the 32nd International Symposium on Computational Geometry*, 2016. doi:10.4230/LIPIcs.SoCG.2016.52.
- 17 Alfredo García Olaverri, Marc Noy, and Javier Tejel. Lower bounds on the number of crossing-free subgraphs of K_n . *Comput. Geom.*, 16(4):211–221, 2000. doi:10.1016/S0925-7721(00)00010-9.
- 18 Daniel Rutschmann. On chains and point configurations with many triangulations. Master’s thesis, ETH Zurich, Zürich, Switzerland, 2021.
- 19 Francisco Santos and Raimund Seidel. A better upper bound on the number of triangulations of a planar point set. *J. Comb. Theory, Ser. A*, 102(1):186–193, 2003. doi:10.1016/S0097-3165(03)00002-5.
- 20 Raimund Seidel. On the number of triangulations of planar point sets. *Comb.*, 18(2):297–299, 1998. doi:10.1007/PL00009823.
- 21 Micha Sharir and Adam Sheffer. Counting triangulations of planar point sets. *Electron. J. Comb.*, 18(1), 2011. URL: http://www.combinatorics.org/Volume_18/Abstracts/v18i1p70.html.
- 22 Micha Sharir and Emo Welzl. Random triangulations of planar point sets. In *Proceedings of the 22nd ACM Symposium on Computational Geometry*, 2006. doi:10.1145/1137856.1137898.
- 23 Warren D. Smith. *Studies in Computational Geometry Motivated by Mesh Generation*. PhD thesis, Princeton University, Princeton, USA, 1989.

Nearly-Doubling Spaces of Persistence Diagrams

Donald R. Sheehy  

Department of Computer Science, North Carolina State University, Raleigh, NC, USA

Siddharth S. Sheth

Department of Computer Science, North Carolina State University, Raleigh, NC, USA

Abstract

The space of persistence diagrams under bottleneck distance is known to have infinite doubling dimension. Because many metric search algorithms and data structures have bounds that depend on the dimension of the search space, the high-dimensionality makes it difficult to analyze and compare asymptotic running times of metric search algorithms on this space.

We introduce the notion of nearly-doubling metrics, those that are Gromov-Hausdorff close to metric spaces of bounded doubling dimension and prove that bounded k -point persistence diagrams are nearly-doubling. This allows us to prove that in some ways, persistence diagrams can be expected to behave like a doubling metric space. We prove our results in great generality, studying a large class of quotient metrics (of which the persistence plane is just one example). We also prove bounds on the dimension of the k -point bottleneck space over such metrics.

The notion of being nearly-doubling in this Gromov-Hausdorff sense is likely of more general interest. Some algorithms that have a dependence on the dimension can be analyzed in terms of the dimension of the nearby metric rather than that of the metric itself. We give a specific example of this phenomenon by analyzing an algorithm to compute metric nets, a useful operation on persistence diagrams.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Topological Data Analysis, Persistence Diagrams, Gromov-Hausdorff Distance

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.60

Funding This research was supported by the NSF under grant CCF-2017980.

1 Introduction

A persistence diagram is a topological summary commonly used in topological data analysis (TDA). Ever since their introduction, persistence diagrams have been a popular tool to compare the shapes of point clouds, metric spaces, and real-valued functions.

A significant advantage of persistence diagrams over many other topological invariants is that they come equipped with a natural metric, the bottleneck distance, and thus topological features are rendered not only qualitative, but also quantitative. This opens the possibility of doing metric analysis on persistence diagrams, such as (approximate) nearest neighbor search or range search.

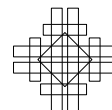
Many metric proximity search algorithms and data structures have asymptotic running time bounds in terms of the doubling dimension of the search space [6, 10]. The metric space of persistence diagrams with the bottleneck distance is known to have infinite doubling dimension [8], making it unclear whether one ought to apply standard data structures such as cover trees [1] or net trees [10] to search in this space. Although the space of all persistence diagrams is infinite-dimensional, all hope is not lost. In this paper, we show that the bottleneck space of bounded persistence diagram (i.e., those whose points are in a bounded region) is close in a Gromov-Hausdorff sense to a finite-dimensional space. Our approach is to consider a very general class of quotient metrics generalizing the persistence plane and then bound the doubling dimension of bottleneck distances over such metrics. We also show that for some algorithms whose running time depends on the doubling dimension,



© Donald R. Sheehy and Siddharth S. Sheth;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 60; pp. 60:1–60:15
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



it can sometimes suffice to be close to a low-dimensional metric in order to achieve similar running times. Specifically, we show how to construct nets efficiently in these so-called nearly-doubling metrics.

As a first attempt at explaining why the bottleneck space of persistence diagrams appears to behave like a low-dimensional space in some experiments (see [16]), one might hope that “real-world” persistence diagrams naturally live in a low-dimensional subspace. Certainly, there are cases where data naturally live on a low-dimensional manifold and zooming in, one sees only the low-dimensional structure. However, this is not true of persistence diagrams. Zooming in can increase rather than decrease the apparent dimension. As a result, the key idea in this paper is not to look for a low-dimensional subspace, but rather a different low-dimensional space that is provably Gromov-Hausdorff close.

2 Related Work

There are numerous examples of metric search algorithms where search performance depends on the underlying space’s doubling dimension. The performance guarantees of navigating nets in [14] depend on an exponential function of the doubling dimension. The same is true for Clarkson’s *sb* data structure [6] and Har-Peled and Mendel’s net-trees [10].

The bottleneck matching data structure of Efrat et al. [7] runs in time $O(n^{1.5} \log^d n)$ in \mathbb{R}^d using ℓ_∞ distance. Kerber et al. [12] apply the geometric intuition of Efrat et al. [7] to the space of persistence diagrams and give the current state-of-the-art algorithm for computing the bottleneck distance between persistence diagrams. The running time is $O(n^{1.5} \log n)$. Kerber and Nigmetov also acknowledge the high dimensionality of some spaces as a problem when they build spanners that minimize distance computations for such spaces [13]. In their work, they explicitly mention persistence diagrams as a motivating example of an expensive to compute metric, but their theoretical results only apply to doubling metrics. Nigmetov [16] gave many experimental results showing that methods geared towards doubling spaces still work well on persistence diagrams. In this paper, we give some indication for why similar results could apply in the (non-doubling) setting of persistence diagrams.

Fasy et al. explore the infinite doubling dimension of persistence diagrams in [8] with a nearest neighbor data structure. They replace the bounded persistence plane with a grid to reduce the doubling dimension of the space of bounded persistence diagrams.

Our approach is based on the fact that the persistence plane is a quotient of the ℓ_∞ plane modulo the diagonal. This approach was first defined by Bubenik and Elchesen [2, 3]. They use this definition of the persistence plane in terms of quotient metrics to prove results on more general spaces of persistence diagrams.

Choudhary and Kerber [5] introduce the idea of a t -restricted doubling dimension where the dimension is computed only by focusing on balls of radius at most t . The notion of nearly-doubling metrics we introduce in this paper takes the opposite approach, capturing the doubling behavior at sufficiently large scales. This is more appropriate for persistence diagrams, because the high-dimensionality is present at arbitrarily small scales.

Huang et al. [11] present a similar result for clustering problems where they compute weighted approximations of subsets of doubling metrics in polynomial time.

3 Definitions

3.1 Metric Spaces

A *metric space*, (X, d) is a set X and a metric d . This is the default metric space used in this paper. The distance between $a \in X$ and a set Y is given by $d(x, Y) := \inf_{b \in Y} d(a, b)$. The *diameter* of a set X is $\text{diam}(X) = \sup_{a, b \in X} d(a, b)$. An *r -ball centered at a* , denoted by $B(a, r)$, is the set of all points in X within distance at most r from a . The *spread* of a finite metric space is the ratio of its diameter to its smallest pairwise distance.

A collection of sets Y *covers* X if the union of the sets in Y contains X . An *r -cover* is a collection of sets of diameter at most $2r$ that covers X . A special case of an r -cover is a cover by metric balls of radius r . A *minimum r -cover* is an r -cover of X of minimum cardinality. The *covering number* of X is $N_r(X) = |Y|$ where Y is a minimum r -cover of X . The *r -metric entropy* of X is defined as $H_r(X) = \log_2 N_r(X)$.

The *doubling dimension* of X , denoted $\text{dim}(X)$, is the minimum number d such that every subset $S \subseteq X$ can be covered by 2^d sets of half the diameter of S . As observed in the original work on doubling dimension [14], a ball in a d -dimensional metric space can be covered with at most 2^{2d} balls of half the radius.¹ If $\text{dim}(X)$ is finite, then X is a *doubling metric*. Throughout this paper, all mentions of dimension refer to the doubling dimension.

3.2 Packings and Coverings

A set $X_r \subset X$ is said to be *r -dense* or an *r -sample* of X if $X \subset \bigcup_{x \in X_r} B(x, r)$. A set $Z \subset X$ is said to be *r -separated* or an *r -packing* of X if $d(z_i, z_j) > r$ for all distinct $z_i, z_j \in Z$. If $Z \subset X$ is both, r -dense and r -separated, then Z is an *r -net* of X .

The *packing number* of a set X , given by $M_r(X)$, is the size of the maximum r -packing of X . The *sampling number* of X , given by $S_r(X)$, is the size of the minimum r -sampling of X . There is a well-known relationship between the packing and covering numbers of a set X known from [15]. We present a proof for completeness.

► **Lemma 1** (Packing-Covering Duality). *If X is a metric set and r is some distance, then,*

$$M_{2r}(X) \leq N_r(X) \leq M_r(X).$$

Proof. For the second inequality, let P be a maximum r -packing of X and $S = \bigcup_{p \in P} B(p, r)$ be such that S is not an r -cover of X . Thus, there exists $y \in X$ such that $d(y, p) > r$ for all $p \in P$. Therefore, P is not a maximum r -packing of X and so $N_r(X) \leq M_r(X)$.

For the first inequality, let $Y = \{Y_1, \dots, Y_N\}$ be an r -cover of X of size $N_r(X)$. Assume there exists P' , a $2r$ -packing of X , of size $N_r(X) + 1$. By the pigeonhole principle there exists Y_i such that two elements of P' , say p, p' , are in Y_i because Y is an r -cover. Thus, $d(p, p') \leq \text{diam}(Y_i) \leq 2r$. Therefore, P' is not a 2ε -packing and so $M_{2r}(X) \leq N_r(X)$. ◀

A similar lemma holds for the covering number and sampling number.

► **Lemma 2.** *If X is a metric set and r is some distance, then*

$$S_{2r}(X) \leq N_r(X) \leq S_r(X).$$

¹ In some prior work, the definition of doubling dimension is given in terms of coverage of metric balls rather than general covers. That definition suffers from several drawbacks; most notably, it is not monotone with respect to subsets.

60:4 Nearly-Doubling Spaces of Persistence Diagrams

Lemma 2 gives us a relationship between the doubling dimension computed using centered and uncentered balls of diameter $2r$.

Krauthgamer and Lee [14] say that the doubling dimension computed by covering a metric ball with balls of half the radius is a 2-approximation of the actual doubling dimension. The following lemma shows that the converse of that statement is also true.

► **Lemma 3.** *Let X be metric space. If, for any $r > 0$, there exists an $r/2$ -sample of a ball $B(x, r)$ in X of cardinality 2^ρ , then $\dim(X) \leq 2\rho$.*

Proof. Let $Z \subset X$ be a set of diameter $2r$. Then $Z \subseteq B(z, 2r)$ for any $z \in Z$. So there exists Z' , an r -sample of $B(z, 2r)$, of cardinality 2^ρ . Moreover, for every $z' \in Z'$ there exists an $r/2$ -sample of cardinality 2^ρ of a ball $B(z', r)$. Therefore, there exists an $r/2$ -sample of Z of cardinality at most $2^{2\rho}$. Thus, from Lemma 2 there exists an $r/2$ -cover of Z of cardinality at most $2^{2\rho}$ and so $\dim(X) \leq 2\rho$. ◀

Krauthgamer and Lee [14] prove that an r -packing of an $O(r)$ -ball has at most $2^{O(d)}$ points. A version of this lemma with more precise constants is the following.

► **Lemma 4 (Standard Packing Lemma).** *If X is a metric space of dimension d and $Z \subset B(x, r)$ for some $x \in X$ is an λ -packing then $|Z| \leq (2\Delta)^d$ where $\Delta \leq \frac{2r}{\lambda}$ is the spread of Z .*

Let X be a metric space and let Y be a subspace. The *quotient metric space* $(X/Y, d_{X/Y})$ is defined so that $d_{X/Y}([a], [b]) := \min\{d(a, b), d(a, Y) + d(b, Y)\}$. There also exists a surjective quotient map, $q : X \rightarrow X/Y$ such that $q(x) = [x]$.

3.3 Bottleneck Distance

Let X be a metric space and let A and B be two finite subsets of the same cardinality. A matching between A and B is bijection $m : A \rightarrow B$. The *bottleneck* of a matching m is

$$\max_{a \in A} d(a, m(a)).$$

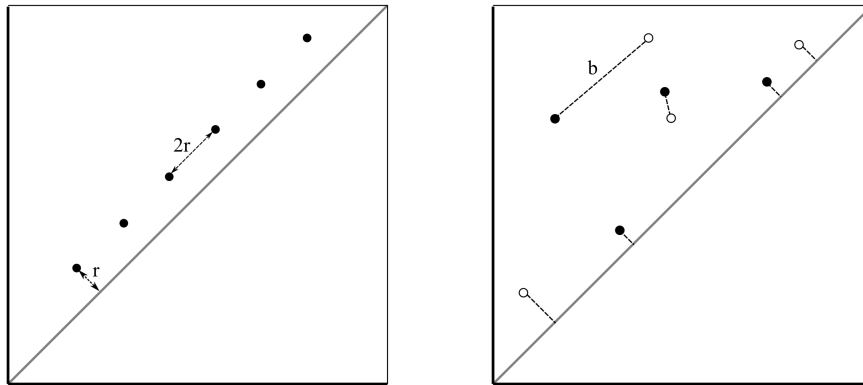
The *bottleneck distance* between A and B is the minimum of the bottleneck over all possible matchings between A and B .

3.4 The Persistence Plane

The *persistence plane* \mathbb{P} is the quotient $(\mathbb{R}^2, \ell_\infty)$ modulo the diagonal $\{(x, x) \mid x \in \mathbb{R}\}$. The point associated with the equivalence class of the diagonal in the persistence plane is called the *diagonal point*. The dimension of \mathbb{P} is infinite as shown in Figure 1a. This means that a quotient of two doubling metric spaces can be infinite-dimensional.

A *persistence diagram* is a multiset of points in the persistence plane. The natural metric on persistence diagrams is the bottleneck distance. To ensure diagrams A and B have the same cardinality, we augment A with $|B|$ copies of the diagonal point and we augment B with $|A|$ copies of the diagonal point. Then the *bottleneck distance for persistence diagrams* is the bottleneck distance between the augmented diagrams.

Treating the persistence plane as a quotient metric is due to Bubenik and Elchesen. [2]. Although this perspective is nonstandard, it provides several significant benefits. It simplifies algorithms for computing bottleneck distance, because having a single “point” representing the entire diagonal allows one to more easily perform augmentation compared to standard approaches [12]. It also simplifies sketching [17], in which one uses an approximate persistence diagram that has fewer distinct points with multiplicity.



(a) The Persistence Plane.

(b) Bottleneck Matching.

■ **Figure 1** (a) shows why the persistence plane has infinite doubling dimension. A ball of radius r centered at the diagonal would contain infinitely many points at distance r from the diagonal but a ball of radius $r/2$ centered off the diagonal can cover only one of them. (b) shows a bottleneck matching between two persistence diagrams.

3.5 Gromov-Hausdorff Distance

Given compact sets A and B in a metric space X , the *Hausdorff distance* between them is

$$d_H(A, B) = \max\{\max_{a \in A} \min_{b \in B} d(a, b), \max_{b \in B} \min_{a \in A} d(a, b)\}.$$

For metric spaces (P, d_P) and (Q, d_Q) , a *correspondence* between P and Q is a relation $\mathcal{R} \subseteq P \times Q$ such that for its canonical projections on P and Q , we have $\pi_P(\mathcal{R}) = P$ and $\pi_Q(\mathcal{R}) = Q$ respectively. The *distortion* of \mathcal{R} is defined as

$$\text{distort}(\mathcal{R}) := \sup_{(p_1, q_1), (p_2, q_2) \in \mathcal{R}} |d_P(p_1, p_2) - d_Q(q_1, q_2)|.$$

The *Gromov-Hausdorff distance*, d_{GH} , is a metric on compact metric spaces [9] defined as

$$d_{GH}(P, Q) := \frac{1}{2} \inf\{\text{distort}(\mathcal{R}) \mid \mathcal{R} \subseteq P \times Q \text{ is a correspondence}\}.$$

In this paper we say two metric spaces are ε -close to mean that the Gromov-Hausdorff distance between them is at most ε . The Gromov-Hausdorff distance is a generalization of the Hausdorff distance in the sense that if P and Q are subsets of a common metric space, then their Gromov-Hausdorff distance is bounded by their Hausdorff distance. So if the Hausdorff distance between two subspaces of a metric space is bounded, the Gromov-Hausdorff distance between them is also bounded.

4 ε -Close Quotient Metric Spaces

A quotient metric space X/Y can have very high (or infinite) dimension even if X and Y are low-dimensional. A perfect example of this phenomenon is the persistence plane, which has infinite dimension despite being the quotient of a 2-dimensional space by a 1-dimensional

60:6 Nearly-Doubling Spaces of Persistence Diagrams

subspace. In this section, we show how to approximate a quotient space with a lower dimensional quotient space. We first present a lemma on the dimension of a quotient of a doubling metric modulo a finite subset.

► **Lemma 5.** *Let X be a d -dimensional metric space. If $Y \subset X$ is finite, then*

$$\dim(X/Y) \leq d + \log_2 |Y|.$$

Proof. Let $S \subseteq X/Y$ be such that $\text{diam}(S) = 2\varepsilon$. Let $q : X \rightarrow X/Y$ be the quotient map. There exists a subset $S' \subseteq X$ such that $q(S') = S$. For $y \in Y$, define the Voronoi cell of y restricted to S' to be

$$\text{Vor}_{|S'}(y) := \{x \in S' \mid d(x, y) = d(x, Y)\}.$$

Then, for each $y \in Y$, we have

$$\begin{aligned} \text{diam}(\text{Vor}_{|S'}(y)) &:= \sup_{a, b \in \text{Vor}_{|S'}(y)} d(a, b) \\ &\leq \sup_{a, b \in \text{Vor}_{|S'}(y)} \min\{d(a, b), d(a, y) + d(b, y)\} \\ &= \sup_{a, b \in \text{Vor}_{|S'}(y)} \min\{d(a, b), d(a, Y) + d(b, Y)\} \\ &= \sup_{a, b \in \text{Vor}_{|S'}(y)} d_{X/Y}(q(a), q(b)) \\ &\leq \sup_{a, b \in S'} d_{X/Y}(q(a), q(b)) \\ &= 2\varepsilon \end{aligned}$$

So, $\text{Vor}_{|S'}(y)$ is a set with diameter 2ε , and, by the definition of doubling dimension, has an $\varepsilon/2$ -cover of size at most 2^d . Let C be the union of these covers for all $y \in Y$. Then C will $\varepsilon/2$ -cover S' in X . Distances only decrease in the quotient, so the sets $\{q(U) \mid U \in C\}$ will $\varepsilon/2$ -cover S in X/Y . So, we have an $\varepsilon/2$ -cover of S of size at most $|Y|2^d$ and thus,

$$\dim(X/Y) \leq \log_2(|Y|2^d) = d + \log_2 |Y|. \quad \blacktriangleleft$$

► **Theorem 6.** *Let X and Y be compact metric spaces such that $Y \subseteq X$ and $\dim(X) = d$. Then, X/Y is ε -close to a metric of dimension at most $d + H_{\varepsilon/2}(Y)$.*

Proof. Let Y_ε be a minimum ε -sample of Y . Then by Lemma 2, the cardinality of the minimum $\varepsilon/2$ -cover of Y is at least $|Y_\varepsilon|$. Therefore, $H_{\varepsilon/2}(Y) \geq \log |Y_\varepsilon|$. So, Lemma 5 implies that X/Y_ε has dimension at most $d + H_{\varepsilon/2}(Y)$. It will suffice to show that $d_{GH}(X/Y, X/Y_\varepsilon) \leq \varepsilon$.

Let $q : X \rightarrow X/Y$ and $q_\varepsilon : X \rightarrow X/Y_\varepsilon$ denote the canonical quotient maps. Let $\mathcal{R} \subseteq X/Y \times X/Y_\varepsilon$ be the relation

$$\mathcal{R} = \{(q(x), q_\varepsilon(x)) \mid x \in X\}.$$

Quotient maps are surjective, so the canonical projections of \mathcal{R} satisfy $\pi_{X/Y}(\mathcal{R}) = X/Y$ and $\pi_{X/Y_\varepsilon}(\mathcal{R}) = X/Y_\varepsilon$. Thus, \mathcal{R} is a correspondence between X/Y and X/Y_ε .

Because Y_ε is an ε -sample of Y , for any $a \in X$, we have

$$d(a, Y) \leq d(a, Y_\varepsilon) \leq d(a, Y) + \varepsilon.$$

It follows that

$$\begin{aligned} d_{X/Y}(q(a), q(b)) &= \min\{d(a, b), d(a, Y) + d(b, Y)\} \\ &\leq \min\{d(a, b), d(a, Y_\varepsilon) + d(b, Y_\varepsilon)\} \\ &= d_{X/Y_\varepsilon}(q_\varepsilon(a), q_\varepsilon(b)), \end{aligned}$$

and also,

$$\begin{aligned} d_{X/Y_\varepsilon}(q_\varepsilon(a), q_\varepsilon(b)) &= \min\{d(a, b), d(a, Y_\varepsilon) + d(b, Y_\varepsilon)\} \\ &\leq \min\{d(a, b), d(a, Y) + d(b, Y) + 2\varepsilon\} \\ &\leq d_{X/Y}(q(a), q(b)) + 2\varepsilon. \end{aligned}$$

We can then bound the distortion of \mathcal{R} as follows.

$$\text{distort}(\mathcal{R}) = \sup_{a, b \in X} |d_{X/Y}(q(a), q(b)) - d_{X/Y_\varepsilon}(q_\varepsilon(a), q_\varepsilon(b))| \leq 2\varepsilon.$$

Because Y is compact, Y_ε is finite and X/Y_ε is the required ε -close space with doubling dimension at most $d + H_{\varepsilon/2}(Y)$. \blacktriangleleft

Note that the preceding theorem does not directly apply to the persistence plane because it is not compact. We resolve this issue in Section 9 using bounded persistence diagrams.

5 Nearly-Doubling Metric Spaces

A metric space X is ε -nearly-doubling if there exists a doubling metric space Y such that $d_{GH}(X, Y) \leq \varepsilon$. In the previous section we showed that quotients of a doubling metric by a compact set are ε -nearly-doubling with a dimension that depends on ε . In later sections, we will show how bottleneck spaces are also nearly doubling with a focus on subsets of persistence diagrams. Before proceeding to those results, we explain the sense in which nearly doubling metrics share some of the properties of doubling metrics. In particular, they can behave like doubling metrics down to scale $O(\varepsilon)$. The most useful fact about doubling metrics is that they satisfy the packing property described in Lemma 4. The following lemma shows how to bound the size of packings of sufficiently large balls in nearly-doubling metrics.

► **Lemma 7 (Nearly-Doubling Packing Lemma).** *Let $r, \lambda \in \mathbb{R}$ be such that $\lambda < r$. Let S be a λ -packing of a ball $B(c, r)$ in a metric space (X, d) . Let (X', d') be a d -dimensional metric space such that $d_{GH}(X, X') \leq \varepsilon$. If $\lambda = \alpha\varepsilon$ for some $\alpha > 2$, then $|X| \leq \left(\frac{2\alpha+2}{\alpha-2}\Delta\right)^d$ where $\Delta \leq \frac{2r}{\lambda}$ is the spread of S .*

Proof. Because $d_{GH}(X, X') \leq \varepsilon$ there exists a correspondence \mathcal{R} between X and X' such that $|d(a, b) - d'(a', b')| \leq 2\varepsilon$ for all $(a, a'), (b, b') \in \mathcal{R}$. For each $x \in S$, choose $f(x) \in X'$ to be a point such that $(x, f(x)) \in \mathcal{R}$. Let $S' = \{f(x) \mid x \in S\}$. For any $a, b \in S$,

$$|d(a, b) - d'(f(a), f(b))| \leq 2\varepsilon.$$

Because S is a λ -packing, we have that for all $a, b \in S$,

$$\begin{aligned} d'(f(a), f(b)) &\geq d(a, b) - 2\varepsilon \\ &\geq \lambda - 2\varepsilon. \end{aligned}$$

60:8 Nearly-Doubling Spaces of Persistence Diagrams

In other words, distinct points of S map to points of distance at least $\lambda - 2\varepsilon > 0$. It follows that f is a bijection and S' is a $(\lambda - 2\varepsilon)$ -packing. The distortion bound on \mathcal{R} implies that

$$\begin{aligned} \text{diam}(S') &= \sup_{a,b \in S} d'(f(a), f(b)) \\ &\leq \sup_{a,b \in S} d(a, b) + 2\varepsilon \\ &= \text{diam}(S) + 2\varepsilon \\ &\leq 2r + 2\varepsilon. \end{aligned}$$

So, the spread Δ' of S' is at most $\frac{2r+2\varepsilon}{\lambda-2\varepsilon}$. Using the fact that $\alpha\varepsilon = \lambda < r$, we get the following bound on Δ' in terms of the spread Δ of S .

$$\Delta' \leq \frac{2r + 2\varepsilon}{\lambda - 2\varepsilon} = \frac{2r + \frac{2\lambda}{\alpha}}{\frac{\alpha-2}{\alpha}\lambda} < \frac{2r\alpha + 2r}{(\alpha - 2)\lambda} \leq \frac{\alpha + 1}{\alpha - 2} \Delta.$$

We then use the fact that f is bijection and apply Lemma 4, to get

$$|S| = |S'| \leq \left(\left(\frac{2\alpha + 2}{\alpha - 2} \right) \Delta \right)^d. \quad \blacktriangleleft$$

The nearly-doubling packing lemma explains why algorithms and data structures defined for doubling metrics work for nearly-doubling metrics down to some scale. We give a specific example and analysis in the following section.

6 Clarkson's Algorithm in Nearly-Doubling Spaces

The main theme of this paper is that although some metric spaces are high-dimensional, they are Gromov-Hausdorff close to low-dimensional metrics. We showed this is true for a wide class of compact quotient metrics in Section 4 and will extend these results to the bottleneck space of bounded persistence diagrams in Section 9. Before we tackle those problems, we will show that being close to a low-dimensional metric has some benefit. In particular, there are basic algorithms for doubling metrics that will also be efficient in nearly-doubling metrics.

In this section we analyze the performance of an algorithm for constructing a λ -net in a nearly-doubling metric space. The main result will be that as long as $\lambda \geq 3\varepsilon$, the running time can be bounded in terms of the dimension of an ε -close metric.

The algorithm we will consider for computing the net is sometimes called Clarkson's Algorithm. It is a variation of an algorithm originally due to Clarkson [6] with some simplifications due to Har-Peled and Mendel [10] and Sheehy [19]. The idea is to produce a net by greedy sampling (also known as farthest point sampling or Gonzalez ordering). Any point may be selected first and each subsequent point maximizes the distance to the points selected so far, stopping when the distance is less than the target scale λ . An open source Python implementation is available [18]. Given a finite subspace P of a doubling metric space X with cardinality n , the algorithm computes a net of P in time $O\left(n \log \frac{\text{diam}(P)}{\lambda}\right)$. The big-O hides terms that are exponential in the dimension, but if the dimension is too high, the simpler upper bound of $O(n^2)$ applies. So, for inputs with polynomial spread in doubling metrics, the running time is $O(n \log n)$. Thus, our goal is to show that similar guarantees hold in nearly-doubling metrics.

The algorithm follows an incremental construction of the greedy sampling. The points in the net will be numbered p_0, p_1, \dots . The first point p_0 is chosen arbitrarily. Let $P_i := \{p_0, \dots, p_{i-1}\}$ be the i th prefix, and $\lambda_i := d(p_i, P_i)$ be the insertion radius of p_i . For every

point $p \in P_i$ the algorithm maintains a list of $q \in P \setminus P_i$ that are the reverse nearest neighbors of p . Essentially, this is the Voronoi cell of p . A *neighbor graph* is defined on the Voronoi cells that is guaranteed to have an edge (p_i, p_j) if adding a point in the Voronoi cell of p_i can affect the Voronoi cell of p_j . At each step i the algorithm has the points of P_i in a max heap with the key of a point p_a given by the distance from p_a to the farthest point in its Voronoi cell. The algorithm simply pops a point p_a from the heap, and adds the farthest point p_i to the net. The Voronoi cells and the neighbor graph are updated. The neighbor graph stores exactly the cells that could change so one only needs to check the Voronoi cells of the neighbors of p_a . New edges in the neighbor graph incident to p_i can be found among the 2-hop neighbors (i.e., neighbors of neighbors) of p_a . A key insight to make the algorithm efficient is to keep some extra edges (p_a, p_b) in the graph as long as $d(p_a, p_b) \leq 3\lambda_i$. Clarkson showed that the desired neighbors will all satisfy such a condition.

► **Theorem 8.** *Let ε and λ be such that $\lambda \geq 3\varepsilon$. If X is ε -close to a d -dimensional metric space, then Clarkson’s algorithm computes a λ -net of X in $2^{O(d)}n \log_2(n \frac{\text{diam}(X)}{\lambda})$ time.*

Proof. There are three aspects of the algorithm that must be analyzed: the update to the neighbor graph, the heap operations, and the update to the Voronoi cells. In the i th iteration, the points P_i form a λ_i -net. So, Lemma 7 and the condition that $d(p_a, p_b) \leq 3\lambda_i$ for neighbors p_b of p_a imply that the degree of p_a is $2^{O(d)}$. This means that updating the neighbor graph takes constant time per point. It also means that the number of keys to update in the heap is constant per iteration. So, the heap operations take $2^{O(d)}n \log_2 n$ time in the worst case.

To analyze the number of distance computations performed when updating the Voronoi cells, we apply an analysis similar to that used by Har-Peled and Mendel [10]. For each point $x \in P$, we want to count how many times we compute the distance from q to the newly inserted point p_i (to see if it should change Voronoi cells). In such cases, we say p_i touches x .

Let $x \in \text{Vor}(p_k)$ be touched by newly inserted point $p_i \in \text{Vor}(p_j)$.

$$\begin{aligned} d(x, p_i) &\leq d(x, p_k) + d(p_k, p_i) \\ &\leq d(x, p_k) + d(p_k, p_j) + d(p_j, p_i) \\ &\leq \lambda_i + 3\lambda_i + \lambda_i = 5\lambda_i. \end{aligned}$$

For an integer m , define the annulus $A_m = \{p_i \mid 2^m \leq \lambda_i < 2^{m+1} \text{ and } p_i \text{ touches } x\}$. If $p_i \in A_m$ then $d(x, p_i) \leq 5\lambda_i \leq 5 \cdot 2^{m+1}$. So $A_m \subset B(x, 5 \cdot 2^{m+1})$. Moreover A_m is 2^m -separated. Therefore, by Lemma 7, $|A_m| \leq 2^{O(d)}$. Thus, x is touched at most a constant number of times in each annulus. The algorithm stops as soon as λ_i is smaller than λ , so, the number of nonempty annuli that can contain x is at most $\log_2 \frac{\text{diam}(P)}{\lambda}$. It follows that the total work of updating the Voronoi cells takes $2^{O(d)}n \log_2 \frac{\text{diam}(P)}{\lambda}$ time.

Combining the running time of the graph update, the heap operations and the cell updates, we get a total running time of $2^{O(d)}n \log_2 \left(n \frac{\text{diam}(P)}{\lambda}\right)$. ◀

7 Bottleneck Metrics

If the doubling dimension of X is d , then a d -dimensional k -point diagram is a set of k elements of X . Let $X^{(k)}$ be the space of k -point diagrams in X with the bottleneck metric.

► **Theorem 9.** *If X is a d -dimensional metric space, then for all integers $k \geq 1$, we have $\dim(X^{(k)}) \leq 4kd$.*

60:10 Nearly-Doubling Spaces of Persistence Diagrams

Proof. Let $D \in X^{(k)}$ and positive $r \in \mathbb{R}$ be chosen arbitrarily. It will suffice to construct an $r/2$ -sample of $B(D, r)$ of size 2^{2kd} . For each point $p_i \in D$, there is an $r/2$ -sample $\{x_{i,j}\}_{j \in [2^{2d}]}$ of $B(p_i, r)$ in X . For $j : [k] \rightarrow [2^{2d}]$, let

$$C_j := \{x_{i,j(i)} \mid i \in [k]\}.$$

Assume all diagrams $A = \{a_i\}_{i \in [k]}$ are indexed so the bottleneck matching with D has a_i matched to p_i . This means that each $a_i \in A$ is in $B(p_i, r)$. If $j(i)$ is the index of the nearest point in the sample of $B(p_i, r)$, then there is a matching $A \rightarrow C_j$ with bottleneck at most $r/2$. So, the set $C = \{C_j \mid j : [k] \rightarrow [2^{2d}]\}$ is an $r/2$ -sample of $B(D, r)$. Clearly, $|C| = 2^{2kd}$, so the dimension of $X^{(k)}$ is at most $2 \log_2(|C|) = 4kd$. ◀

If the bottleneck space is over a quotient metric, then Lemma 5 and Theorem 9 together yield the following corollary.

► **Corollary 10.** *Let X/Y be a quotient metric induced by a finite subspace Y over X . Then, $\dim(X/Y^{(k)}) \leq 4k(d + \log_2 |Y|)$.*

For many metric spaces such as ℓ_p -spaces, maximal sets with a fixed diameter are metric balls. In such metrics, or if the doubling dimension is defined in terms of metric balls (as opposed to general covers), there is no need for the factor of 4 in the dimension for the preceding two results. In particular this holds in the persistence plane.

For bottleneck spaces defined over nearly doubling metrics, it is useful to have the following theorem showing that the mapping from metric spaces to bottleneck spaces is Lipschitz.

► **Theorem 11.** *If X and Y are compact metric spaces, then for all integers $k \geq 1$,*

$$d_{GH}(X^{(k)}, Y^{(k)}) \leq d_{GH}(X, Y).$$

Proof. Let \mathcal{R} be a minimum distortion correspondence between X and Y . Let 2ε be the distortion of \mathcal{R} . Let $[k] = \{0, \dots, k-1\}$. Let $\mathcal{R}^{(k)}$ denote the correspondence between $X^{(k)}$ and $Y^{(k)}$ defined as

$$\mathcal{R}^{(k)} = \{(\{a_i\}_{i \in [k]}, \{b_i\}_{i \in [k]}) \mid \exists \text{ bijection } m : [k] \rightarrow [k] \text{ s.t. } \forall i, (a_i, b_{m(i)}) \in \mathcal{R}\}.$$

To show that $d_{GH}(X^{(k)}, Y^{(k)}) \leq \varepsilon$, it is sufficient to bound the distortion of $\mathcal{R}^{(k)}$.

Let (A, B) and (A', B') be arbitrary pairs in the $\mathcal{R}^{(k)}$, where $A = \{a_i\}_{i \in [k]}$, $A' = \{a'_i\}_{i \in [k]}$, $B = \{b_i\}_{i \in [k]}$, and $B' = \{b'_i\}_{i \in [k]}$. Without loss of generality, we may assume they are indexed so that for all j , we have $(a_j, b_j) \in \mathcal{R}$ and $(a'_j, b'_j) \in \mathcal{R}$. Let $\eta : [k] \rightarrow [k]$ be the permutation of indices that gives the bottleneck matching between A and A' , i.e.,

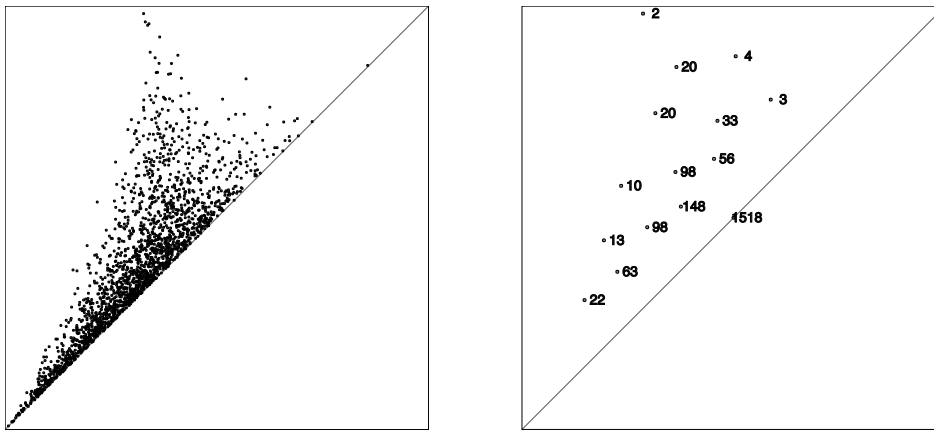
$$d_B(A, A') = \max_{i \in [k]} d_X(a_i, a'_{\eta(i)}).$$

It follows that

$$\begin{aligned} d_B(B, B') &\leq \max_{j \in [k]} d_Y(b_j, b'_{\eta(j)}) \\ &\leq \max_{j \in [k]} (d_X(a_j, a'_{\eta(j)}) + 2\varepsilon) \\ &= d_B(A, A') + 2\varepsilon. \end{aligned}$$

Symmetrically, we have $d_B(A, A') \leq d_B(B, B') + 2\varepsilon$ and thus, $\text{distort}(\mathcal{R}^{(k)}) \leq 2\varepsilon = \text{distort}(\mathcal{R})$. To conclude, we observe that

$$d_{GH}(X^{(k)}, Y^{(k)}) \leq \frac{1}{2} \text{distort}(\mathcal{R}^{(k)}) \leq \frac{1}{2} \text{distort}(\mathcal{R}) = d_{GH}(X, Y). \quad \blacktriangleleft$$



■ **Figure 2** The image on the left shows a persistence diagram for points sampled on a sphere. The image on the right shows a sketch of that persistence diagram with first 14 points. The number to the right of each point shows its multiplicity in the sketch.

8 Bottleneck Spaces with Multiplicity

A k -point diagram D with multiplicity is a set $\underline{D} \subseteq X$ of cardinality k and a function $m_D : \underline{D} \rightarrow \mathbb{Z}_+$. The *total multiplicity* of D is $m_D = \sum_{p \in \underline{D}} m_D(p)$. In this section, we consider the space $X^{(k,N)}$ of k -point diagrams with total multiplicity N . This may be viewed as a subset of $X^{(N)}$, consisting of those diagrams with at most k distinct points. In Theorem 12, we show that $X^{(k,N)}$ has a dimension that depends only logarithmically on N .

The motivation for studying such diagrams with multiplicity again comes from persistence diagrams. It often happens that points in a persistence diagram have multiplicity. Recently, it was shown that actively seeking such multiplicity can lead to efficient sketches of persistence diagrams [17].

A simple sketching algorithm is to run Clarkson's Algorithm (see Section 6) on a persistence diagram starting with the diagonal point until k points have been added. The algorithm maintains the Voronoi cells of the points in the net and therefore one simply sets the multiplicity of each point to be the number of points in its Voronoi cell. The result is a k -point sketch, D_k , of a diagram D . It is then straightforward to show that $d_B(D, D_k)$ is at most $d_H(\underline{D}, \underline{D}_k)$ [17]. The advantage of the sketch is that it is a guaranteed approximation and can be represented in much less size. In some cases (i.e., for $k = O(\log n)$) it is asymptotically faster to compute the bottleneck distance between sketches than the full diagrams. There is nothing special about persistence diagrams in this algorithm. An example of a sketch is shown in Figure 2.

If $D \in X^{(N)}$, then $D_k \in X^{(k,N)}$. Theorem 9 gives a bound of $4Nd$ on the dimension of $X^{(N)}$. However, as we show in the theorem below, the sketch will live in a lower dimensional space.

► **Theorem 12.** *Let X be a d -dimensional metric space. If k and N are positive integers such that $k \leq N$, then $\dim(X^{(k,N)}) \leq \min\{4Nd, 2k(2d + \log_2(2Nk))\}$*

Proof. Let $C \in X^{(k,N)}$ and $r \in \mathbb{R}$ be with $r > 0$ be chosen arbitrarily. We will construct an $r/2$ -cover of $B(C, r)$ in $X^{(k,N)}$ by constructing an $r/2$ -sample as follows. For each $p \in \underline{C}$ there exists an $r/2$ -sample U_p of $B(p, r)$ of size at most 2^{2d} . This means that if $d(x, p) \leq r$, then for some $u_i \in U_p$, we have $d(x, u_i) \leq r/2$.

60:12 Nearly-Doubling Spaces of Persistence Diagrams

Let $\mathcal{U} = \cup_{p \in \underline{C}} U_p$. Because $|\underline{C}| = k$, we know that $|\mathcal{U}| \leq 2^{2d}k$. Let $S \subseteq X^{(k,N)}$ be defined as

$$S := \{D \mid \underline{D} \subset \mathcal{U}, |\underline{D}| \leq k, m_D = N\}.$$

For S to be an $r/2$ -sample of $B(C, r)$ we will show that for all $E \in B(C, r)$ there exists $D \in S$ such that $d_B(D, E) \leq r/2$. Let $E = (\underline{E}, m_E)$ be any diagram in $B(C, r)$. For every $q \in \underline{E}$, there exists $p \in \underline{C}$ such that $d(p, q) \leq r$. So, there exists $q' \in U_p$ such that $d(q, u_i) \leq r/2$ for some $u_i \in U_p$.

Consider a diagram $D = (\underline{D}, m_D)$ where $\underline{D} = \{q' \mid q \in \underline{E}\}$ and $m_D(q') = m_E(q)$ for all $q' \in \underline{D}$. By construction $D \in S$. The bottleneck distance is bounded as follows

$$d_B(D, E) \leq \max_{q \in \underline{E}} d(q, q') \leq r/2.$$

It follows that S is an $r/2$ -sample.

We bound the size of S as follows. Because $|\mathcal{U}| \leq 2^{2d}k$, there are at most $\binom{2^{2d}k}{k} \leq k^k 2^{2kd}$ different choices of \underline{D} for a diagram in S . The number of ways to distribute multiplicity N over the k points of \underline{D} is $\binom{N+k-1}{k-1} \leq (2N)^k$, because $N \geq k$. It then follows that

$$|S| \leq k^k 2^{2kd} (2N)^k = (2Nk 2^{2d})^k.$$

So, the doubling dimension is at most

$$2 \log_2(|S|) \leq 2 \log_2(2Nk 2^{2d})^k = 2k(2d + \log_2(2Nk)).$$

On the other hand, treating the diagram as a collection of N points without multiplicity and applying the bounds for diagrams without multiplicity (Theorem 9) yields a dimension at most $4Nd$. Combining these two upper bounds on the dimension completes the proof. \blacktriangleleft

9 The Space of Bounded Persistence Diagrams

From the preceding two sections we get an approximation of single-class quotient spaces and a bound on the doubling dimension of finite point bottleneck spaces respectively. These results come together in the space of bounded persistence diagrams to form a nearly low dimensional subspace of persistence diagrams.

The persistence plane is denoted by $P = (\mathbb{R}^2, \ell_\infty) / \{(x, x) \mid x \in \mathbb{R}\}$. Let P_0 denote the bounded persistence plane obtained by restricting P to $[0, 1] \times [0, 1]$. Then, $P_0^{(N)}$ is the bottleneck space of N -point *bounded* persistence diagrams.

The key to finding low-dimensional spaces near $P_0^{(N)}$ is to first find a low-dimensional space near the persistence plane. Theorem 6 gives a recipe for doing so. There is an ε -sample of the diagonal of the bounded persistence plane of size $\lceil \frac{1}{2\varepsilon} \rceil$. So, one can consider the plane modulo the ε -sample rather than modulo the whole diagonal. The resulting metric space is denoted P_ε . It is a special case of the construction in Theorem 6, and thus the following lemma is immediate.

► **Lemma 13.** For all $\varepsilon > 0$, $\dim(P_\varepsilon) \leq 2 + \log_2 \lceil \frac{1}{2\varepsilon} \rceil$ and $d_{GH}(P_0, P_\varepsilon) \leq \varepsilon$.

► **Theorem 14.** The bottleneck space of N -point bounded persistence diagrams, $P_0^{(N)}$ is ε -close to a space of dimension at most $4N(2 + \log_2 \lceil \frac{1}{2\varepsilon} \rceil)$.

Proof. By Theorem 12 and Lemma 13,

$$\dim(\mathbb{P}_\varepsilon^{(N)}) \leq 4N \dim(\mathbb{P}_\varepsilon) \leq 4N(2 + \log_2 \left\lceil \frac{1}{2\varepsilon} \right\rceil).$$

Moreover, Theorem 11 implies that

$$d_{GH}(\mathbb{P}_0^{(N)}, \mathbb{P}_\varepsilon^{(N)}) \leq d_{GH}(\mathbb{P}_0, \mathbb{P}_\varepsilon) \leq \varepsilon. \quad \blacktriangleleft$$

Thus, the space of bounded N -point persistence diagrams is nearly low-dimensional. We can further lower the dimension of the space using sketching. Having fewer points with multiplicity decreases the dimension.

► **Lemma 15.** For all positive integers N, k such that $N \geq k$,

$$d_{GH}(\mathbb{P}_0^{(N)}, \mathbb{P}_0^{(k,N)}) \leq \sqrt{\frac{1}{2k}}.$$

Proof. Given an N -point diagram D , the greedy sketching algorithm produces a k -point diagram D_k with mass N . The bottleneck distance is well-defined for all persistence diagrams, so it will suffice to bound the Hausdorff distance. As $\mathbb{P}_0^{(k,N)}$ is a subspace of $\mathbb{P}_0^{(N)}$, the Hausdorff distance will be the maximum of $d_B(D, D_k)$ over all bounded N -point persistence diagrams. The greedy sketch produces for each k , an ε_k -net of \underline{D} with multiplicities so that $d_B(D, D_k) = \varepsilon_k$. The maximum size of an ε_k -net in \mathbb{P}_0 restricted to the region above the diagonal is $\frac{1}{2\varepsilon_k^2}$. It follows that $k \leq \frac{1}{2\varepsilon_k^2}$ and therefore, $\varepsilon_k \leq \sqrt{\frac{1}{2k}}$. So, for all bounded N -point persistence diagrams D , we have $d_B(D, D_k) \leq \sqrt{\frac{1}{2k}}$ and so the Gromov-Hausdorff distance bound follows. \blacktriangleleft

We can now combine the previous results to prove the following theorem.

► **Theorem 16.** The space $\mathbb{P}_0^{(N)}$ of bounded N -point persistence diagrams is $(\varepsilon + \sqrt{\frac{1}{2k}})$ -close to a metric of dimension at most $2k(4 + 2 \log_2 \lceil \frac{1}{2\varepsilon} \rceil + \log_2(2Nk))$.

Proof. First, the triangle inequality, Lemma 15, and Theorem 6 that

$$d_{GH}(\mathbb{P}_0^{(N)}, \mathbb{P}_\varepsilon^{(k,N)}) \leq d_{GH}(\mathbb{P}_0^{(N)}, \mathbb{P}_0^{(k,N)}) + d_{GH}(\mathbb{P}_0^{(k,N)}, \mathbb{P}_\varepsilon^{(k,N)}) \leq \sqrt{\frac{1}{2k}} + \varepsilon.$$

Then, Theorem 12 and Lemma 13 implies

$$\begin{aligned} \dim(\mathbb{P}_\varepsilon^{(k,N)}) &\leq 2k(2 \dim(\mathbb{P}_\varepsilon) + \log_2(2Nk)) \\ &\leq 2k(2 \dim(\mathbb{P}_\varepsilon) + \log_2(2Nk)) \\ &\leq 2k(4 + 2 \log_2 \left\lceil \frac{1}{2\varepsilon} \right\rceil + \log_2(2Nk)). \end{aligned} \quad \blacktriangleleft$$

10 Conclusion

In this paper, we analyze several generalizations of metric spaces that arise naturally in topological data analysis, with the goal of bounding their dimension. Although the most significant of these, the bottleneck distance for persistence diagrams is infinite-dimensional, we show that in an important sense, it can behave like a low-dimensional space.

The idea of analyzing the running time of an algorithm in terms of the dimension of a nearby metric leads to many natural questions. For example, it should be possible to build linear-size spanners with ε (additive) slack if the input is ε -close to a doubling metric by a direct application of the ideas from Section 6. It is interesting to ask what other metric constructions that are known to be efficient in doubling metrics are also efficient in nearly-doubling metrics.

Although our general study of bottleneck spaces over quotient metrics was primarily motivated by the special case of persistence diagrams, this is not the only example. Other methods in topological data analysis produce different quotient metrics of the type studied in this paper, for example in the work of Carrière and Oudot on Mapper [4]. It remains to find more such examples. It also remains to consider more general quotient metrics, i.e., those defined by an arbitrary equivalence relation rather than just a subset.

Lastly, the results of this paper imply that in many cases, one could hope that metric analysis on collections of persistence diagrams is a reasonable thing to do. Not only will the entropy of the collection be bounded, many standard algorithms designed for doubling metrics should work well without change.

References

- 1 Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, pages 97–104, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1143844.1143857.
- 2 Peter Bubenik and Alex Elchesen. Universality of persistence diagrams and the bottleneck and wasserstein distances, 2021. arXiv:1912.02563.
- 3 Peter Bubenik and Alex Elchesen. Virtual persistence diagrams, signed measures, wasserstein distances, and banach spaces, 2021. arXiv:2012.10514.
- 4 Mathieu Carrière and Steve Oudot. Structure and stability of the 1-dimensional mapper. In *SoCG*, 2016.
- 5 Aruni Choudhary and Michael Kerber. Local doubling dimension of point sets. In *CCCG 2015 Proceedings*, 2015.
- 6 Kenneth L. Clarkson. Nearest neighbor searching in metric spaces: Experimental results for ‘sb(s)’. Preliminary version presented at ALENEX99, 2003.
- 7 A. Efrat, A. Itai, and M. J. Katz. Geometry Helps in Bottleneck Matching and Related Problems. *Algorithmica*, 31(1):1–28, September 2001. doi:10.1007/s00453-001-0016-8.
- 8 Brittany Terese Fasy, Xiaozhou He, Zhihui Liu, Samuel Micka, David L. Millman, and Binhai Zhu. Approximate Nearest Neighbors in the Space of Persistence Diagrams. *arXiv:1812.11257 [cs]*, March 2021. arXiv:1812.11257.
- 9 Misha Gromov. *Metric Structure for Riemannian and Non-Riemannian Spaces*. Birkhauser, 1999.
- 10 Sariel Har-Peled and Manor Mendel. Fast Construction of Nets in Low-Dimensional Metrics and Their Applications. *SIAM Journal on Computing*, 35(5):1148–1184, January 2006. doi:10.1137/S0097539704446281.
- 11 Lingxiao Huang, Shaofeng H.-C. Jiang, Jian Li, and Xuan Wu. Epsilon-coresets for clustering (with outliers) in doubling metrics. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 814–825, 2018. doi:10.1109/FOCS.2018.00082.
- 12 Michael Kerber, Dmitriy Morozov, and Arnur Nigmatov. Geometry Helps to Compare Persistence Diagrams. *ACM Journal of Experimental Algorithmics*, 22:1–20, December 2017. doi:10.1145/3064175.

- 13 Michael Kerber and Arnur Nigmatov. Metric spaces with expensive distances. *International Journal of Computational Geometry and Applications*, 30(02):141–165, June 2020. doi: 10.1142/S0218195920500077.
- 14 Robert Krauthgamer and James R. Lee. Navigating nets: Simple algorithms for proximity search. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '04*, pages 798–807, USA, 2004. Society for Industrial and Applied Mathematics.
- 15 G. G. Lorentz. Metric entropy and approximation. *Bulletin of the American Mathematical Society*, 72(6):903–937, 1966. doi:bams/1183528486.
- 16 Arnur Nigmatov. *Comparison of Topological Summaries*. PhD thesis, TU Graz, 2019.
- 17 Don Sheehy and Siddharth Sheth. Sketching persistence diagrams. In *SoCG*, 2021.
- 18 Donald R. Sheehy. greedypermutations, 2020. URL: <https://github.com/donsheehy/greedypermutation>.
- 19 Donald R. Sheehy. One hop greedy permutations. In *Proceedings of the 32nd Canadian Conference on Computational Geometry*, pages 221–225, 2020.

From Geometry to Topology: Inverse Theorems for Distributed Persistence

Elchanan Solomon ✉ 🏠 

Department of Mathematics, Duke University, Durham, NC, USA

Alexander Wagner ✉ 

Department of Mathematics, Duke University, Durham, NC, USA

Paul Bendich

Department of Mathematics, Duke University, Durham, NC, USA

Geometric Data Analytics, Inc., Durham, NC, USA

Abstract

What is the “right” topological invariant of a large point cloud X ? Prior research has focused on estimating the full persistence diagram of X , a quantity that is very expensive to compute, unstable to outliers, and far from injective. We therefore propose that, in many cases, the collection of persistence diagrams of many small subsets of X is a better invariant. This invariant, which we call “distributed persistence,” is *perfectly parallelizable*, more stable to outliers, and has a rich inverse theory. The map from the space of metric spaces (with the quasi-isometry distance) to the space of distributed persistence invariants (with the Hausdorff-Bottleneck distance) is globally bi-Lipschitz. This is a much stronger property than simply being injective, as it implies that the inverse image of a small neighborhood is a small neighborhood, and is to our knowledge the only result of its kind in the TDA literature. Moreover, the inverse Lipschitz constant depends on the size of the subsets taken, so that as the size of these subsets goes from small to large, the invariant interpolates between a purely geometric one and a topological one. Lastly, we note that our inverse results do not actually require considering all subsets of a fixed size (an enormous collection), but a relatively small collection satisfying simple covering properties. These theoretical results are complemented by synthetic experiments demonstrating the use of distributed persistence in practice.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology

Keywords and phrases Applied Topology, Persistent Homology, Inverse Problems, Subsampling

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.61

Related Version *Full Version*: <https://arxiv.org/abs/2101.12288>

Supplementary Material *Software (Source Code)*: <https://github.com/aywagner/DIPOLE>

Funding *Elchanan Solomon*: AFOSR Grant FA9550-18-1-0266.

Alexander Wagner: NSF CCF-1934964

Paul Bendich: AFOSR Grant FA9550-18-1-0266.

1 Introduction

Morphometric techniques in data analysis can be loosely divided into the geometric and the topological. Geometric techniques, like landmarks, the Gromov-Hausdorff metric, optimal transport methods, PCA, MDS [21], LLE [31], and Isomap [34], are designed to capture some combination of global and local metric structure. Many geometric methods can be solved exactly or approximately via spectral methods, and hence are fast to implement using iterative and sketching algorithms. In contrast, topological techniques, like t-SNE [36], UMAP [25], Mapper [33], and persistent homology, aim to capture large-scale connectivity structure in data. The growing popularity of t-SNE and UMAP as dimensionality reduction methods suggests that many data sets are topologically, but not metrically, low-dimensional. In this paper, we introduce a new technique into topological data analysis (TDA) that:



© Elchanan Solomon, Alexander Wagner, and Paul Bendich;
licensed under Creative Commons License CC-BY 4.0

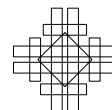
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 61; pp. 61:1–61:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1. Provably interpolates between topological and geometric structure (Theorem 20).
2. Is *perfectly parallelizable*.
3. Is provably stable to perturbation of the data (Proposition 5).
4. Is provably invertible, with globally stable inverse (Theorems 14, 20, 25, and Corollary 23).
Moreover, these inverse results do not require computing the full invariant, but a relatively small subset that can largely be chosen at random (Propositions 27 and 28).
5. Suggests new methods for a host of morphometric challenges, ranging from dimensionality reduction to feature extraction (Section 6).

The theoretical guarantees provided here are, to our knowledge, unmatched by any other method in topological data analysis. In addition to these theoretical contributions, we demonstrate our theoretical results empirically on synthetic data sets.

2 The Distributed Topology Problem

Let λ be an invariant of metric spaces (X, d_X) . For $k \in \mathbb{Z}$, we can define a distributed invariant Λ_k that maps the metric space X to the set of pairs $\{(S, \lambda(S)) \mid S \subset X, |S| = k\}$ if $k > 0$ and to \emptyset otherwise. Put another way, $\Lambda_k(X)$ records the values of λ on subsets of X of a fixed size.

When the computational complexity of λ scales poorly in the size of X , it is much faster to compute λ for many small subsets of X . Λ_k takes this intuition to its limit by performing this calculation for *all* subsets of a given size. Although it is unfeasible to actually compute Λ_k in its entirety, sampling from Λ_k is simple. This distinguishes Λ_k from the original invariant λ , which, in general, cannot be “sampled from” or broken into smaller pieces. Moreover, Λ_k may contain just as much, if not more, information than λ :

- Let λ send a finite point cloud X in \mathbb{R}^d to its Euclidean distance matrix. For all $k \geq 2$, Λ_k contains the same information as λ .
- Let λ send a finite point cloud X in \mathbb{R}^d to its diameter. For any $k \geq 2$, Λ_k can be used to deduce λ .
- Let λ send a finite point cloud X in \mathbb{R}^d to its mean. For any $k \geq 1$, Λ_k can be used to deduce λ . In fact, if $k < |X|$, Λ_k determines X up to rigid motion.

Finally, Λ_k is more robust than λ , as outliers in X have no impact on Λ_k for outlier-free subsets $S \subset X$. The theoretical goal of this paper is to address the following questions:

► **Problem 1.** *If λ is a topological invariant of metric spaces, how much information is contained in Λ_k for various k ? Does Λ_k determine λ , or perhaps contain strictly more information?*

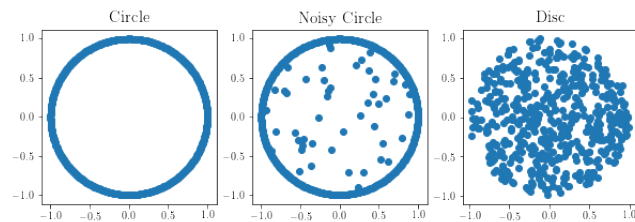
► **Problem 2.** *How does the information contained in Λ_k depend on the parameter k ?*

► **Problem 3.** *What information can be deduced from Λ_k if we can only compute it for a relatively small collection of subsets?*

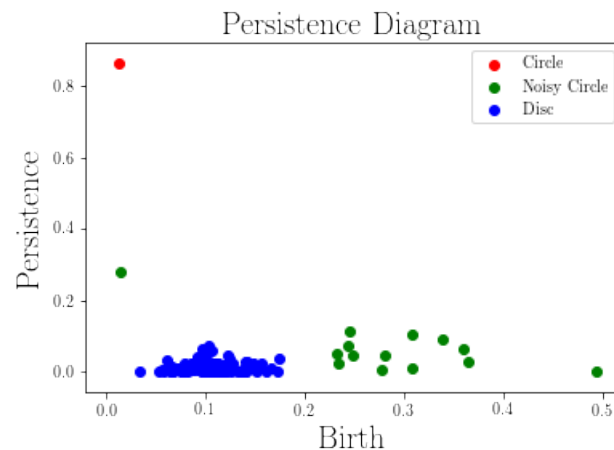
2.1 Case Study: The Noisy Circle

To illustrate the advantage of working with distributed invariants, we compare three data sets of 500 points. The first is spaced regularly around a circle, the second sampled uniformly from the unit disc, and the third contains 450 points on the circle and 50 points sampled from the disc (we call this the *noisy circle*), see Figure 1. For each of these point clouds, we compute their full 1-dimensional persistence diagrams, see Figure 2. In addition, for each

point cloud, we sample 1000 subsets of size 10, compute the resulting 1000 1-dimensional persistence diagrams, vectorize them as *persistence images*¹, and average the results, see Figure 3. The persistence diagram of the noisy circle is most similar to that of the disc (in Bottleneck distance), demonstrating that ordinary persistence does not see the circle around which most of the data points are clustered. The distributed persistence, however, tells a different story. The distribution for the noisy circle interpolates between the distributions of the other two spaces, but is substantially closer to that of the circle than the disc.



■ **Figure 1** Three point clouds: the circle, the noisy circle, and the disc.

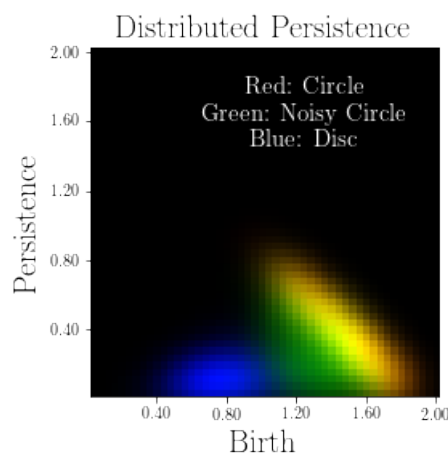


■ **Figure 2** The persistence diagrams of our point clouds, plotted in birth-persistence coordinates.

3 Prior Work on Distributed Topology

In [9], Chazal et al. propose the following framework. Given a metric measure space (\mathbb{X}, ρ, μ) , sample m points and compute the persistence landscape of the associated Vietoris-Rips filtration. This procedure produces a random persistence landscape, λ , whose distribution is denoted Ψ_μ^m . Repeating this procedure n times and averaging produces the empirical average landscape, an unbiased estimator of the average landscape $E_{\Psi_\mu^m}[\lambda]$. This approach is similar to the distributed topological invariants considered in this paper, except we consider a collection of topological invariants as a labeled set rather than taking their sum. Though

¹ This is a technique for turning a persistence diagram into a function by placing a Gaussian kernel at each dot in the persistence diagram, with mean and variance varying by location, cf. [1].



■ **Figure 3** Averaged distributed persistence images of our three spaces. The dominant orange/yellow region is the overlay of the circle (red) distribution and the noisy circle (green) distribution.

Bubenik [3] gives conditions in Theorem 5.11 under which a collection of persistence diagrams may be reconstructed from the average of their corresponding persistence landscapes, such an inverse exists only generically, and is highly unstable.

The main theorem of [9] is that the average landscape is stable with respect to the underlying measure. Specifically, if μ and ν are two probability measures on the same metric space (\mathbb{X}, ρ) , the sup norm between induced average landscapes is bounded by $m^{1/p}W_{\rho,p}(\mu, \nu)$ for any $p \geq 1$. Similar results were obtained in [2] for distributions of persistence diagrams of subsamples. In particular, Blumberg et al. showed that the distribution of barcodes with the Prokhorov metric is stable with respect to the associated compact metric measure space in the Gromov-Prokhorov metric. Both results are analogous to the stability of the distributed topological invariants given in Proposition 5. However, working with labeled collections of distributed topological invariants, we are also able to provide inverse stability results, such as our main Theorem 20, which states that changes in the metric structure are bounded with respect to changes in the distributed topological invariants.

In [26], Mémoli developed the study of *curvature sets*, an invariant introduced by Gromov that consists of computing the distance matrices of every subset of a fixed size in an ambient metric space. Shortly after this paper appeared, Gómez and Mémoli [18] released a manuscript studying the collection of persistence diagrams of subsets of bounded cardinality in an ambient metric space. This construction is similar to ours, with the following key differences: firstly, we take subsets of a fixed cardinality k , or else cardinalities in a small neighborhood of k , whereas Gómez and Mémoli consider all subsets of cardinality at most k , and, secondly, we have different conventions for which homological degrees to compute. More importantly, that paper differs from ours in the nature of the results: Gómez and Mémoli are focused on computing this invariant for simple spaces, and giving examples of when their invariant characterizes the homotopy type of the underlying space. This paper is focused on inverse results of a geometric flavor, trying to understand how distributed topological invariants characterize the quasi-isometry type of the underlying space.

In [4], Bubenik et al. consider unit disks, denoted D_K , of surfaces of constant Gaussian curvature K with $K \in [-2, 2]$. Since these spaces are all contractible, their reduced singular homology is trivial and global homology cannot distinguish them. However, the authors prove that the maximum Čech persistence for three points sampled from D_K determines K .

The authors also successfully apply the same empirical framework of average persistence landscapes from [9] to experimentally determine the curvature of D_K for various K . The authors in [14] used average persistence landscapes to provide experimental verification of a known phase transition. Finally, the authors in [24] use average persistence landscapes to achieve improved results, compared to standard machine learning algorithms, in disease phenotype prediction based on subject gene expressions.

4 Background

The content of this paper assumes familiarity with the concepts and tools of persistent homology. Interested readers can consult the articles of Carlsson [6] and Ghrist [16] and the textbooks of Edelsbrunner and Harer [15] and Oudot [30]. We include the following primer for readers interested in a high-level, non-technical summary.

Persistent homology records the way topology evolves in a parametrized sequence of spaces. To apply persistent homology to a metric space, a pre-processing step is needed that converts the metric space into such a sequence. The two classical ways of doing this are called the Rips and Čech filtrations, respectively; the former is much easier to compute than the latter, but contains less geometric information. Both consist of inserting simplices into the metric space at a parameter value equal to the proximity of the associated vertex points. As the sequence of spaces evolves, the addition of certain edges or higher-dimensional simplices changes the homological type of the space – these simplices are called critical. Persistent homology records the parameter values at which critical simplices appear, notes the dimension in which the homology changes, and pairs critical values by matching the critical value at which a new homological feature appears to the critical value at which it disappears. This information is organized into a structure called a persistence diagram, and there are a number of metrics with which persistence diagrams can be compared.

If one forgets about the pairing and retains only the dimension information of the critical values, the resulting invariant is called a Betti curve. Betti curves are simpler to compute and work with than persistence diagrams, but are less informative and harder to compare. Finally, if one also drops the dimension information by taking the alternating sum of the Betti curves, one gets an Euler curve. Euler curves are even less discriminative than Betti curves, but enjoy the special symmetry properties of the Euler characteristic. These symmetries will be put to good use in this paper.

Persistence theory guarantees that a small modification to the parametrization of a sequence of spaces implies only small changes in its persistence diagram. To be precise, if the appearance time of any given simplex is not delayed or advanced by more than ϵ , the persistence diagram as a whole is not distorted by more than ϵ in the appropriate metric (called the *Bottleneck distance*). Throughout this paper we will use the trick of modifying filtrations by rounding their critical values to a fixed, discrete set.

As a rule, the map sending a metric space to its persistence diagram is not injective, as many different point clouds share the same persistence diagram [11, 23, 22]. Moreover, the set of metric spaces sharing a common persistence diagram need not be bounded, so that arbitrarily distinct spaces might have the same persistence. There are a number of constructions in the TDA literature that attempt to correct this lack of injectivity by constructing more sophisticated invariants; these are often called *topological transforms*. Examples include the Persistent Homology Transform [35, 17, 12, 20] and Intrinsic Persistent Homology Transform [29]; consult [28] for a survey of inverse results in persistence. These methods are largely infeasible to approximate, unstable, and provide no global Lipschitz

bounds on their inverse, so two wildly different spaces may produce arbitrarily similar (though not exactly identical) transforms. The distributed topology invariant studied in this paper is injective, easy to sample from, stable, and with Lipschitz inverse.

We conclude with an analysis of the computational complexity of persistence calculations. Persistence calculations are $O(N^\omega)$, where N is the number of simplices in the complex and ω is the matrix multiplication constant [27]. For a metric space X , the number of $(d+1)$ -dimensional simplices in the Rips complex is $\binom{|X|}{d+2}$, which are needed for computing persistence in degree d . Thus the computational complexity is $O(\binom{|X|}{d+2}^\omega)$, which is huge even for small values of d . Computing persistence of M k -element subsets is $O(M \binom{k}{d+2}^\omega)$, which is orders of magnitude smaller for the values of M used in the experiments of Section 7.

5 Theoretical Results

In what follows, we let λ be any of the following four topological invariants: (1) Rips Persistence (RP), (2) Rips Euler Curve (RE), (3) Čech Persistence (CP), and (4) Čech Euler Curve (CE). To be precise, RP and CP consist of persistence diagrams for every homological degree. When working with either of these invariants, the Bottleneck or Wasserstein distance is the maximum of the Bottleneck or Wasserstein distances over all degrees. Our decision to focus on these four invariants is motivated by a desire to keep the following analysis as simple and concrete as possible, and many of the arguments and theorems below carry through, with minor modification, for other invariants. Indeed, a very similar analysis works for functional persistence, where the sampling consists of picking k points at random and computing functional persistence on their induced subcomplex; details of this proof will appear in future work.

5.1 Stability

A result of the following form is standard in the TDA literature, and demonstrates the ease of producing stable invariants using persistent homology.

► **Definition 4.** Let (X, d_X) and (Y, d_Y) be metric spaces. A map $\phi : (X, d_X) \rightarrow (Y, d_Y)$ is an ϵ -quasi-isometry if $|d_X(x_1, x_2) - d_Y(\phi(x_1), \phi(x_2))| \leq \epsilon$ for all $x_1, x_2 \in X$. The quasi-isometry distance between X and Y is the smallest ϵ for which such a map exists.

► **Proposition 5.** Let $\phi : (X, d_X) \rightarrow (Y, d_Y)$ be an ϵ -quasi-isometry of metric spaces. Then for all subsets $S \subseteq X$, and λ either RP or CP, $d_B(\lambda(S), \lambda(\phi(S))) \leq \epsilon$, where d_B is the Bottleneck distance on persistence diagrams.

Proof. This follows immediately from the Gromov-Hausdorff stability theorem for persistence diagrams of metric spaces [8, 10]. ◀

5.2 Injectivity

In this section, we show how distributed persistence can be used to reconstruct the isometry type of a metric space. This provides an answer to Problem 1. To help motivate this result, we consider the simple cases of $k = 2$ and $k = 3$.

► **Lemma 6.** For all of our invariants, Λ_2 determines the isometry type of X , and hence also $\lambda(X)$.

Proof. For each invariant, the distance between two points $x, y \in X$ can be read off of $\lambda(x, y)$, so Λ_2 records all the pairwise distances between points in X , and hence the metric d_X . The metric then determines the Rips or Čech complex of X as an abstract metric space. When considering the Čech complex of a point cloud X in Euclidean space, the metric determines the embedding of X up to Euclidean isometry (see [32]), and hence the Čech filtration. ◀

Setting $k = 3$ is sufficient to break the implication of an isometry.

▶ **Lemma 7.** Λ_3 does not determine the isometry type of X .

Proof. A simple counterexample suffices. Let X consist of the vertices of an obtuse triangle with angle $\theta > \pi/2$. Varying the angle θ in $(\pi/2, \pi)$ alters the isometry type of X , but leaves its persistent homology unchanged. ◀

To obtain stronger results, we introduce the following two generalizations, one to the notion of distributivity, and the other to the invariants λ .

▶ **Definition 8.** Let $\mathbf{k} = \{k_1, k_2, \dots, k_r\}$ be a set of positive integers. We write $\Lambda_{\mathbf{k}}$ for the union $\bigcup_{i=1}^r \Lambda_{k_i}$.

▶ **Definition 9.** For any of our four invariants λ , let λ^m be the invariant restricted to the m -skeleton of the Rips or Čech complex, and define $\Lambda_{\mathbf{k}}^m$ analogously.

Setting $m = 0$ provides information only on the cardinality of X . The 1-skeleton contains both geometric and topological information, and its persistence is fast to compute. As m increases, computational complexity goes up, and the resulting invariants record higher-dimensional topological information. The following results demonstrate how knowing sufficiently many Euler characteristic invariants allows one to determine new ones.

▶ **Definition 10.** For a set X , let $K(X)$ be the full simplicial complex on X , which is abstractly equal to the power set of X . A function $f : K(X) \rightarrow \mathbb{R}$ on the simplices of K is called monotone if $f(\sigma) \leq f(\tau)$ when σ is a face of τ . For a subcomplex $T \subseteq K(X)$, we write $T(r)$ to denote the r -sublevel set of f on T .

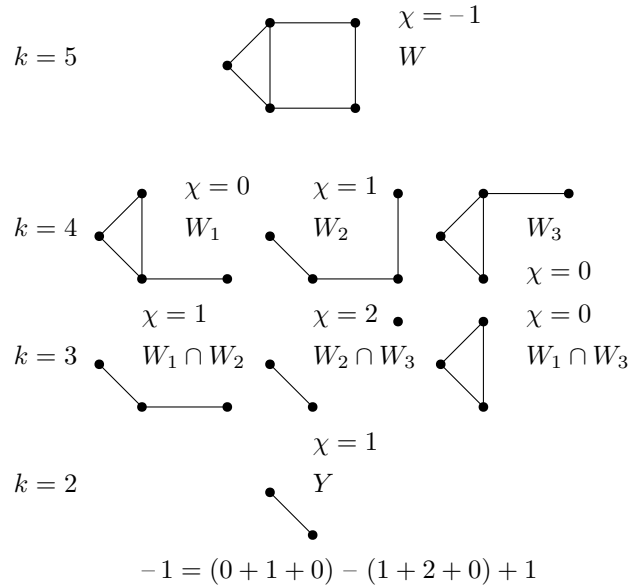
▶ **Lemma 11.** Let R, T_1, \dots, T_n be subcomplexes of $K(X)$, the full complex on X . Writing S^m to denote the m -skeleton of a subcomplex S , suppose that $R^m = \bigcup_{i=1}^n T_i^m$. For $f : K(X) \rightarrow \mathbb{R}$ a monotone function on K , we have:

$$\begin{aligned} \chi(R^m(r)) &= \chi\left(\bigcup_i T_i^m(r)\right) \\ &= \sum_i \chi(T_i^m(r)) \\ &\quad - \sum_{i < j} \chi(T_i^m(r) \cap T_j^m(r)) \\ &\quad + \sum_{i < j < k} \chi(T_i^m(r) \cap T_j^m(r) \cap T_k^m(r)) \quad \dots \\ &\quad + (-1)^{n-1} \chi(T_1^m(r) \cap \dots \cap T_n^m(r)). \end{aligned}$$

Proof. This follows from the inclusion-exclusion property of the Euler characteristic. ◀

▶ **Lemma 12.** Let λ be RE or CE. For any metric space X and $k \geq m + 2$, let $\mathbf{k} = \{k, k - 1, \dots, k - m - 1\}$. Then $\Lambda_{\mathbf{k}}^m$ determines Λ_{k-m-2}^m .

Proof. Let $Y \subset X$ be a subset of size $(k - m - 2)$. Let $\{x_1, \dots, x_{m+2}\}$ be points in $X \setminus Y$, and set $W = Y \cup \{x_1, \dots, x_{m+2}\}$ and $W_i = W \setminus \{x_i\}$. Let $f : K(X) \rightarrow \mathbb{R}$ be the function giving rise to the Rips or Čech filtration, and let $R = K(W)$ and $T_i = K(W_i)$. By construction, $R^m = \bigcup_{i=1}^n T_i^m$, since every $(m + 1)$ -element subset of W lies in some W_i , so we may apply Lemma 11. This gives a formula for the Euler characteristic of $R^m(r)$ in terms of the Euler characteristics of the $T_i^m(r)$ and their intersections. By hypothesis, we know the Euler characteristics of every term in this equation other than the final term, $\chi(T_1^m(r) \cap \dots \cap T_n^m(r)) = \chi(K^m(Y)(r))$, since every other term involves the Euler characteristic of a set with cardinality in \mathbf{k} . This means that we can solve for $\chi(K^m(Y)(r))$ in terms of known quantities, and hence deduce the Euler curve for the Rips or Čech filtration on Y . See Figure 4 for a concrete example. ◀



■ **Figure 4** Our goal is to deduce the Euler Characteristic (at a fixed scale r) of Y , a 1-simplex consisting of $k = 2$ points. This can be derived from the Euler Characteristics of the other subcomplexes in the diagram above.

► **Corollary 13.** *Let λ be RE or CE. For any metric space X and $k \geq m + 2$, let $\mathbf{k} = \{k, k - 1, \dots, k - m - 1\}$. Then $\Lambda_{\mathbf{k}}^m$ determines Λ_2^m .*

Proof. Lemma 12 shows that $\{\Lambda_k^m, \Lambda_{k-1}^m, \dots, \Lambda_{k-m-1}^m\}$ determines Λ_{k-m-2}^m . By the same logic, $\{\Lambda_{k-1}^m, \Lambda_{k-2}^m, \dots, \Lambda_{k-m-2}^m\}$ determines Λ_{k-m-3}^m . Repeating this argument, we can deduce Λ_2^m . ◀

Leveraging Lemma 12, we prove that all of our persistence invariants determine the isometry type of X .

► **Theorem 14.** *For any of the four invariants λ and $k \geq m + 2 > 2$, let $\mathbf{k} = \{k, k - 1, \dots, k - m - 1\}$. Then $\Lambda_{\mathbf{k}}^m$ determines the isometry type of X .*

Proof. When $m \geq 1$, the m -skeleton contains all edges in X , so Lemma 6 applies. If the set $\{k, k - 1, \dots, k - m - 1\}$ contains 2, this follows from Lemma 6. Otherwise, let us assume λ is either RE or CE, as RP or CP contain more information than their Euler characteristic counterparts. By Corollary 13, we can determine Λ_2^m and then apply Lemma 6. ◀

► **Remark 15.** Note that $m = 1$ suffices to apply the prior theorem. As m gets larger, more topological information is needed to determine the isometry type of the underlying space.

5.3 Inverse Stability

We now consider what happens if two metric spaces have distributed invariants which are similar but not identical. We show that this implies a quasi-isometry between X and Y , with constant depending quadratically on the subset size parameter k . This provides a precise answer to Problem 2 on how the distributed invariant interpolates between geometry and topology.

The key insight in the proof of this result is that there is always a way to modify the Rips or Čech filtrations on X and Y to force their distributed invariants to coincide exactly. Taken together with the telescoping trick of Corollary 13, this modified invariant must agree for all subsets of size two. Persistence stability allows us to assert that the modified invariant and the original persistence invariant are a bounded distance apart, so equality of the modified invariant gives near-equality of the Rips or Čech persistences on subsets of size two, which is nothing more than pairwise distance data.

The proposed modification to our filtration consists of rounding it to a discrete set of values. The following technical lemma shows how to pick a rounding set R that aligns two sets of real values without moving any value more than a bounded amount. The proof of this lemma can be found in the full version of the paper.

► **Lemma 16 (Rounding Lemma).** *Let $P = \{p_1 \leq p_2 \leq \dots \leq p_N\}$ and $Q = \{q_1, q_2, \dots, q_N\}$ be two multisets of real numbers. Define $d_i = |p_i - q_i|$, let $\epsilon = \max d_i$ and $\delta = \sum_{i=1}^n d_i$. Then there exists a subset $R \subset \mathbb{R}$ and a map $\pi : P \cup Q \rightarrow R$ sending a real value x to the unique closest element in R (rounding up at midpoints), with:*

1. $\pi(p_i) = \pi(q_i)$ for all i .
2. $|\pi(x) - x| \leq 3\epsilon + 4\delta$.

In particular, since $\epsilon \leq \delta$, we can replace (2) with (2) $|\pi(x) - x| \leq 7\delta$.*

This lemma is central to the proof of the central result of this section, Theorem 20, the details of which can be found in the full version. The preceding definitions clarify the statement of the theorem:

► **Definition 17.** *Let $m < k$ be natural numbers. We define the following partial sum of binomial coefficients:*

$$S(k, m) = \binom{k}{2} + \binom{k}{3} + \dots + \binom{k}{m+1}.$$

► **Definition 18.** *Let (K, f) be a filtered simplicial complex, i.e. a simplicial complex K with a monotone function $f : K \rightarrow \mathbb{R}$ encoding the appearance times of simplices. Given a subset $R \subset \mathbb{R}$, rounding this filtration to R consists of post-composing f with the map sending every element of \mathbb{R} to its nearest element in R (rounding up at midpoints).*

► **Remark 19.** The appearance time of simplices in an R -rounded filtration occur only at values contained in R . The effect of this rounding on the resulting persistence diagrams is to round the birth and death times of its constituent dots; no new points are introduced.

► **Theorem 20.** *Let λ be either RP or CP, and take $k > m > 0$. Let Z and Y be metric spaces, $\phi : Z \rightarrow Y$ a map of sets, and $X \subseteq Z$ a subspace such that $\phi_X : X \rightarrow Y$ is a surjection. Let $\Gamma \subset P(Z)$ be a collection of subsets of cardinality k through $k - m - 1$ satisfying the following two properties:*

- (*X-Covering property*) For every pair of points $\{x_1, x_2\}$ in X there is a subset $S \in \Gamma$ such that $|S| = k$ and $\{x_1, x_2\} \subset S$.
- (*Closure property*) If $S \in \Gamma$ has $|S| = k$, and $S' \subset S$ has $|S'| \geq k - m - 1$, then $S' \in \Gamma$. Suppose that $d_B(\lambda^m(S), \lambda^m(\phi(S))) \leq \epsilon$ for all $S \in \Gamma$. If λ is RP, ϕ_X is a $112k^2\epsilon$ quasi-isometry, and if λ is CP, ϕ_X is a $224S(k, m)\epsilon$ quasi-isometry.

► **Remark 21.** The collection Γ of all subsets of size k through $k - m - 1$ enjoys the covering and closure properties above. However, it is easy to find much smaller collections satisfying the conditions of Theorem 20, see Section 5.5.

► **Remark 22.** Theorem 20 answers Problem 2 by showing that smaller values of k give more control of quasi-isometry type than larger values. This justifies our claim that distributed topology interpolates between local geometry and global topology.

One can shrink the collection Γ further by asking only that its elements cover sufficiently close approximations for X and Y ; in this case, the resulting bound is not in the quasi-isometry distance but in the Gromov-Hausdorff distance.

► **Corollary 23.** Let λ be either RP or CP, and take $k > m > 0$. Let $\phi : Z \rightarrow Y$ be a map of metric spaces, $X \subset Z$ a subspace, and $X' \subset Z$ another, potentially much smaller, subspace with $d_{GH}(X, X') < \delta$. Suppose also that $d_{GH}(\phi(X'), Y) < \delta$. Finally, let $\Gamma \subset P(Z)$ be a collection of subsets of cardinality k through $k - m - 1$ satisfying the covering and closure properties for X' , and such that $d_B(\lambda^m(S), \lambda^m(\phi(S))) \leq \epsilon$ for all $S \in \Gamma$. If λ is RP, then $d_{GH}(X, Y) \leq 112k^2\epsilon + 2\delta$, and if λ is CP, then $d_{GH}(X, Y) \leq 224S(k, m)k^{m+1}\epsilon + 2\delta$.

Proof. Theorem 20 implies that ϕ is a quasi-isometry from X' to $\phi(X')$. We can turn this into a Gromov-Hausdorff matching between X and Y using the facts that $d_{GH}(X, X') < \delta$ and $d_{GH}(\phi(X'), Y) < \delta$, and two applications of the triangle inequality increase the bound by 2δ . ◀

► **Corollary 24.** If $X \subset \mathbb{R}^{d_1}$ and $Y \subset \mathbb{R}^{d_2}$, then the quasi-isometry bound for Čech persistence in Theorem 20 can be replaced with:

$$112k^2 \left(\epsilon + \sqrt{\frac{2d_1}{d_1 + 1}} + \sqrt{\frac{2d_2}{d_2 + 1}} \right)$$

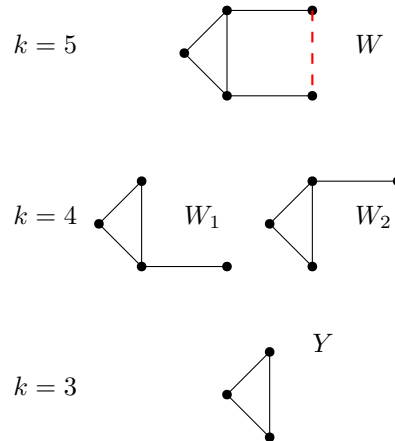
Note that the added terms sum at most to $2\sqrt{2}$, so that this bound is better than the bound given in Theorem 20 for large values of ϵ , but does fail to go to 0 as $\epsilon \rightarrow 0$.

Proof. The Rips and Čech persistence of point clouds in \mathbb{R}^d are always within $\sqrt{\frac{2d}{d+1}}$ of one another in the bottleneck distance, cf. Theorem 2.5 in [13]. The result then follows by replacing Čech persistence with Rips persistence and using the triangle inequality. ◀

5.4 Topology + Sparse Geometry

Our goal now is improve the results of the prior section by giving quasi-isometry bounds that scale linearly in k , rather than quadratically. This can be accomplished by using an inclusion-exclusion argument on the 1-skeleton persistence of X that uses only subsets of size k and $k - 1$, and does not need subsets of size $k - 2$. Namely, given a subset $Y \subset X$ with $|Y| = (k - 2)$, we take $Y = W_1 \cap W_2$ for $|W_1| = |W_2| = (k - 1)$ and $W = (W_1 \cup W_2)$ with $|W| = k$, as shown in Figure 5, and attempt to deduce the Euler characteristic of Y from those of W_1, W_2 , and W . However, the union of the 1-skeleton complexes on W_1 and

W_2 is not the 1-skeleton complex on W , owing to the fact that W contains an extra edge connecting the pair of vertices in $W \setminus Y$. Indeed, this is why we chose to cover W with *three* subsets of cardinality $k - 1$ in Lemma 11.



■ **Figure 5** Our goal is to deduce the Euler Characteristic (at a fixed scale r) of Y , a subcomplex of size $k = 3$, using subcomplexes of size $k = 4$ and $k = 5$. However, the inclusion-exclusion argument fails because the union of the complexes of W_1 and W_2 is not the complex on $W = W_1 \cup W_2$, and the missing edge is shown in red.

The effect of this extra edge on persistence is quite subtle, but its effect on the Euler curve is trivial, as it amounts to subtracting a step function supported on $[r, \infty)$, where r is the appearance time of the extra edge in the complex. If we knew r , we could correct the deficit in our inclusion-exclusion argument. Note that we have the freedom to choose W_1 and W_2 as we like, so to make this argument work we need only know the length of a single edge in X that does not intersect Y . A very small collection of edge lengths suffice to patch up the inclusion-exclusion argument for all subsets of X of size at most k . The following theorem improves on the bounds in Theorem 20 by assuming that ϕ is already known to be a quasi-isometry on a sparse subset $L \subset Z$. The proof can be found in the full version.

► **Theorem 25.** *Let λ, ϕ, Z, Y , and X be as in the statement of Theorem 20, and let $k > m = 1$. Let $L \subset Z$ be a subset satisfying the following geometric condition:*

$$\sum_{(x_i, x_j) \in L \times L} \left| \|x_i - x_j\| - \|\phi(x_i) - \phi(x_j)\| \right| \leq \epsilon_2.$$

Let $\Gamma \subset P(Z)$ be a collection of subsets of cardinality in $\{k, k - 1\}$ satisfying the following two properties:

- *((X, L) -Covering property) For every pair of points $\{x_1, x_2\}$ in X , not both in L , there is a subset $S \in \Gamma$ such that $|S| = k$, $\{x_1, x_2\} \subset S$, and $S \setminus \{x_1, x_2\} \subset L$.*
- *(Closure property) If $S \in \Gamma$ has $|S| = k$, and $S' \subset S$ has $|S'| = k - 1$, then $S' \in \Gamma$.*

Finally, suppose that $d_B(\lambda^1(S), \lambda^1(\phi(S))) \leq \epsilon_1$ for all $S \in \Gamma$. Then ϕ_X is a $56(k + 1)\epsilon_1 + 28\epsilon_2$ quasi-isometry.

► **Remark 26.** Relatively few subsets of cardinality k are needed to satisfy the (X, L) -covering property, as one subset is needed for every pair of points in $(X \setminus L)$, of which there are $\binom{|X \setminus L|}{2}$, and $S = L \cup \{x\}$ works to cover all pairs of the form (l, x) for $l \in L$ and $x \in X$, adding $|X \setminus L|$ more subsets. Finally, to satisfy the closure property, we include all $(k - 1)$ -element subsets of these sets, which multiplies the total number of subsets by at most $(k + 1)$.

5.5 Probabilistic Results

Theorems 20 and 25 and Corollary 23 tell us that we do not need to consider all $\binom{|X|}{k} + \binom{|X|}{k-1} + \dots + \binom{|X|}{k-m-1}$ subsets $S \subseteq X$ of size $|S| \in \{k, \dots, k-m-1\}$, so long as the collection Γ of subsets considered satisfies appropriate cover and closure properties. This still leaves the question of how to produce such a collection Γ in practice. Of the two conditions, the covering property is the more flexible, as the closure property explicitly requires the full downward closure of the appropriate cardinalities. The aim of this section is to show that a relatively small collection of randomly chosen size- k subsets are likely to satisfy the covering property, and hence generate a collection Γ that is both covering and closed. We will assume that $Z = X$ in the language of Theorem 20, i.e. that we are randomly sampling from the space we wish to cover. All proofs can be found in the full version.

The following two propositions, with $p = 2$, provide a lower bound on the probability that a random collection of M subsets covers pairs in X .

► **Proposition 27.** *Let X be a set of size n , and choose M subsets $\{S_1, \dots, S_M\}$ of size k by uniform sampling without replacement. Let $p \leq k$ and A be the outcome that every set of p points (x_1, \dots, x_p) is contained in at least one S_i . Then*

$$P(A) \geq 1 - \binom{n}{p} \left(1 - \left(\frac{k-p+1}{n-p+1}\right)^p\right)^M.$$

► **Proposition 28.** *Let A be as in the prior proposition. For any $\epsilon \in (0, 1)$, if*

$$M \geq (p \log \left(\frac{ne}{p}\right) - \log(1 - \epsilon)) \left(\frac{n-p+1}{k-p+1}\right)^p$$

then $P(A) \geq \epsilon$.

These bounds are further improved in the setting of Corollary 23, when $\{S_1, \dots, S_M\}$ need not cover all pairs of points in X , but all pairs of points in *some* δ -GH approximation X' of X , as there are typically many such approximates with many fewer points than X .

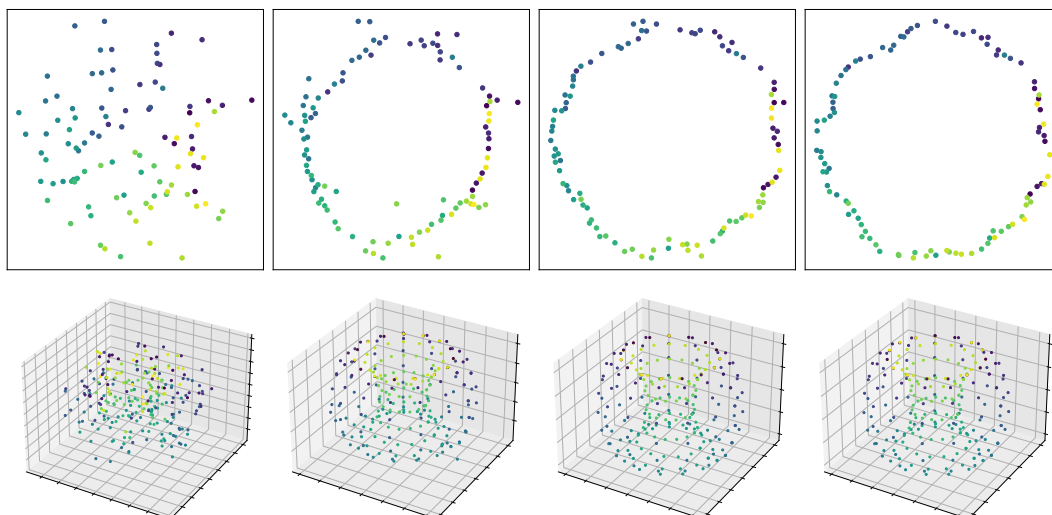
6 Applications

Distributed persistence has myriad applications in machine learning and data analysis, in that it can be applied in many of the same settings as standard persistent homology. We list here a few noteworthy examples.

- (Dimensionality Reduction) Given a high-dimensional data set, the goal of dimensionality reduction is to embed it in lower-dimensional space while preserving its shape. We can force the embedding to preserve the topology of the data by computing a loss comparing the persistence diagrams of many random subsets in the high-dimensional space and in the embedding.
- (Shape Registration) Given two embedded point clouds X and Y modeling the same shape, it can be of interest to learn a map $f : X \rightarrow Y$ aligning corresponding points. Using distributed topology, we can ask for f to preserve the persistence diagrams of many random small subsets of X .
- (Feature Extraction) Given a metric space X , we can compute the persistence diagrams of many random small subsets of X , and, throwing away the subset labelings, obtain a bag-of-persistence-diagrams feature. These features can then be used in machine learning applications.

7 Experiments

The goal of the experiments below is to corroborate the theoretical results in this paper by demonstrating that a loss function built on distributed persistence alone, and sampled on a small number of random subsets, suffices to reconstruct simple metric spaces. Suppose X and Y are finite subsets of Euclidean spaces and $\phi : X \rightarrow Y$ is a surjection. Theorem 20 shows that we may test if ϕ is a quasi-isometry by evaluating $d_B(\lambda^m(S), \lambda^m(\phi(S)))$ for a certain collection of subsets $S \subseteq X$. If X is fixed and Y is variable, we can minimize $d_B(\lambda^m(S), \lambda^m(\phi(S)))$ thanks to the differentiability of persistence computations; this has the effect of bringing Y closer in alignment with X . In the following two synthetic experiments, we follow the methodology described above for X as (1) 100 points evenly distributed on a circle in \mathbb{R}^2 and (2) 256 points evenly distributed on a torus in \mathbb{R}^3 . The codomain Y is initialized to be X with independent Gaussian noise added coordinate-wise. Our aim is to see whether minimizing a distributed topological functional via gradient descent succeeds in correcting for the large geometric distortion of adding Gaussian noise. In both cases, every iteration step consists of uniformly sampling $k = 25$ points, denoted S , from X and taking a step (i.e. perturbing Y) to minimize the loss $W_2^2(D_0(S), D_0(\phi(S))) + W_2^2(D_1(S), D_1(\phi(S)))$, where D_i is the degree i persistence diagram of the Rips filtration. Because we are updating Y based on only a single sample S , we use the Adam optimizer [19] to benefit from momentum. The results of these two experiments can be found in Figure 6, with the first row showing the circle experiment and the second row the torus experiment. For the first (resp. second) row, the first column shows the initial state of Y , and the following columns show Y after successive multiples of 2^{11} (resp. 2^{15}) iterations. For both experiments, we observe the codomain space Y re-organizing itself to closely resemble X . The coloring of the points in Figure 6 denotes their labeling in X , so that points with similar colors are nearby in X . The fact that the color gradients in the final positions of Y are largely continuous affirm that our optimization fixes not only the global geometry of Y , but also the labeled pairwise distances, and hence gives a space quasi-isometric to X . The code used to generate these experiments is available at <https://github.com/aywagner/DIPOLE>.



■ **Figure 6** Synthetic optimization experiments. Columns correspond to initial, intermediate, and final positions of Y . Color denotes labeling.

These experiments are a proof-of-concept but can be developed into a full pipeline for dimensionality reduction. That line of investigation is beyond the scope of this paper, and was carried out by the authors in a separate paper, cf. [38]. A key insight in [38] is that adding a local metric term to the topological loss results in dramatically faster convergence to high-quality embeddings.

8 Conclusion

It has long been understood that computational complexity and sensitivity to outliers are major challenges in the application of persistent homology in data analysis. Moreover, the lack of a stable inverse makes it hard to interpret which geometric information is retained in the persistence diagram, and which is forgotten. Multiple lines of research have sought to address these problems by constructing more sophisticated topological invariants and tools, such as the persistent homology transform, multiparameter persistence, distributed persistence calculations [39], and discrete Morse theory. However, any gains in invertibility are compromised by sizeable increases in computational complexity.

The focus of this paper was the simplest scheme for speeding up persistence calculations: subsampling. Subsampling and bootstrapping are ubiquitous in machine learning and are already being applied in topological data analysis. What we have shown is that this simple approach also enjoys uniquely strong theoretical guarantees. In particular, the manner in which distributed persistence interpolates between geometry and topology is explicitly given by quadratic bounds. Moreover, these theoretical guarantees are complemented by the success that subsampling has seen in the TDA literature, and the robust synthetic experiments shown above.

There remain a number of outstanding problems, both theoretical and computational, that would complement the results of this paper and facilitate its practical application.

- Distributed persistence, as we have defined it, consists of pairs of subsets and persistence diagrams. In many applications, we may wish to take only the persistence diagrams and forget the subset labels. What injectivity results can be obtained in this unstructured setting?
- Individual persistence diagrams can be challenging to work with, due to the fact that the space of diagrams admits no Hilbert space structure [7, 5, 37], though there are a number of effective vectorizations in the literature. How can these be extended or adapted to provide vectorizations of sets of persistence diagrams coming from subsamples of a fixed point cloud? This is a more structured problem than working with arbitrary collections of persistence diagrams.
- If we are interested in recovering the global topology of Euclidean point clouds rather than their quasi-isometry or Gromov-Hausdorff type, it suffices to estimate pairwise distances between points in adjacent Voronoi cells, at least when working with the full Rips or Čech complex and not a skeleton. A careful analysis of this setting could dramatically decrease the Lipschitz constants appearing in Theorem 20.

References

- 1 Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1):218–252, 2017.

- 2 Andrew J. Blumberg, Itamar Gal, Michael A. Mandell, and Matthew Pancia. Robust statistics, hypothesis testing, and confidence intervals for persistent homology on metric measure spaces. *Foundations of Computational Mathematics*, 14(4):745–789, 2014. doi:10.1007/s10208-014-9201-4.
- 3 Peter Bubenik. The persistence landscape and some of its properties. In Nils A. Baas, Gunnar E. Carlsson, Gereon Quick, Markus Szymik, and Marius Thauale, editors, *Topological Data Analysis*, pages 97–117, Cham, 2020. Springer International Publishing.
- 4 Peter Bubenik, Michael Hull, Dhruv Patel, and Benjamin Whittle. Persistent homology detects curvature. *Inverse Problems*, 36(2):025008, January 2020. doi:10.1088/1361-6420/ab4ac0.
- 5 Peter Bubenik and Alexander Wagner. Embeddings of persistence diagrams into hilbert spaces. *Journal of Applied and Computational Topology*, 4(3):339–351, 2020. doi:10.1007/s41468-020-00056-w.
- 6 Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.
- 7 Mathieu Carrière and Ulrich Bauer. On the metric distortion of embedding persistence diagrams into separable Hilbert spaces. In *35th International Symposium on Computational Geometry*, volume 129 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Art. No. 21, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019. URL: <https://mathscinet.ams.org/mathscinet-getitem?mr=3968607>.
- 8 Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*. Springer, 2016.
- 9 Frédéric Chazal, Brittany Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Subsampling methods for persistent homology. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2143–2151, Lille, France, 07–09 July 2015. PMLR. URL: <http://proceedings.mlr.press/v37/chazal15.html>.
- 10 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & computational geometry*, 37(1):103–120, 2007.
- 11 Justin Curry. The fiber of the persistence map for functions on the interval. *Journal of Applied and Computational Topology*, 2(3):301–321, 2018.
- 12 Justin Curry, Sayan Mukherjee, and Katharine Turner. How many directions determine a shape and other sufficiency results for two topological transforms. *arXiv preprint*, 2018. arXiv:1805.09782.
- 13 Vin de Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358, 2007.
- 14 Irene Donato, Matteo Gori, Marco Pettini, Giovanni Petri, Sarah De Nigris, Roberto Franzosi, and Francesco Vaccarino. Persistent homology analysis of phase transitions. *Phys. Rev. E*, 93:052138, May 2016. doi:10.1103/PhysRevE.93.052138.
- 15 Herbert Edelsbrunner and John Harer. *Computational Topology: an Introduction*. American Mathematical Society, 2010.
- 16 Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.
- 17 Robert Ghrist, Rachel Levanger, and Huy Mai. Persistent homology and euler integral transforms. *Journal of Applied and Computational Topology*, 2(1):55–60, 2018.
- 18 Mario Gómez and Facundo Mémoli. Curvature sets over persistence diagrams. *arXiv preprint*, 2021. arXiv:2103.04470.
- 19 Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, 2014. arXiv:1412.6980.
- 20 Henry Kirveslahti and Sayan Mukherjee. Representing fields without correspondences: the lifted euler characteristic transform. *arXiv preprint*, 2021. arXiv:2111.04788.
- 21 J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964. doi:10.1007/BF02289565.

- 22 Jacob Leygonie and Gregory Henselman-Petrusek. Algorithmic reconstruction of the fiber of persistent homology on cell complexes. *arXiv preprint*, 2021. [arXiv:2110.14676](https://arxiv.org/abs/2110.14676).
- 23 Jacob Leygonie and Ulrike Tillmann. The fiber of persistent homology for simplicial complexes. *arXiv preprint*, 2021. [arXiv:2104.01372](https://arxiv.org/abs/2104.01372).
- 24 Sayan Mandal, Aldo Guzmán-Sáenz, Niina Haiminen, Saugata Basu, and Laxmi Parida. A topological data analysis approach on predicting phenotypes from gene expression data. In Carlos Martín-Vide, Miguel A. Vega-Rodríguez, and Travis Wheeler, editors, *Algorithms for Computational Biology*, pages 178–187, Cham, 2020. Springer International Publishing.
- 25 Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint*, 2018. [arXiv:1802.03426](https://arxiv.org/abs/1802.03426).
- 26 Facundo Mémoli. Some properties of gromov–hausdorff distances. *Discrete & Computational Geometry*, 48(2):416–440, 2012.
- 27 Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the twenty-seventh Annual Symposium on Computational Geometry*, pages 216–225, 2011.
- 28 Steve Oudot and Elchanan Solomon. Inverse problems in topological persistence. In *Topological Data Analysis*, pages 405–433. Springer, 2020.
- 29 Steve Oudot and Elchanan Solomon. Barcode embeddings for metric graphs. *Algebraic & Geometric Topology*, 21(3):1209–1266, 2021. [doi:10.2140/agt.2021.21.1209](https://doi.org/10.2140/agt.2021.21.1209).
- 30 Steve Y Oudot. *Persistence theory: from quiver representations to data analysis*, volume 209. American Mathematical Society Providence, 2015.
- 31 Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. [doi:10.1126/science.290.5500.2323](https://doi.org/10.1126/science.290.5500.2323).
- 32 IJ Schoenberg. Remarks to maurice fréchet’s article “sur la définition axiomatique d’une classe d’espace distanciés vectoriellement applicable sur l’espace de hilbert”. *Ann. of Math*, 36:724–732, 1935.
- 33 Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, editors, *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 2007. [doi:10.2312/SPBG/SPBG07/091-100](https://doi.org/10.2312/SPBG/SPBG07/091-100).
- 34 Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. [doi:10.1126/science.290.5500.2319](https://doi.org/10.1126/science.290.5500.2319).
- 35 Katharine Turner, Sayan Mukherjee, and Doug M Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.
- 36 Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL: <http://jmlr.org/papers/v9/vandermaaten08a.html>.
- 37 Alexander Wagner. Nonembeddability of persistence diagrams with $p > 2$ wasserstein metric. *Proceedings of the American Mathematical Society*, 149(6):2673–2677, 2021.
- 38 Alexander Wagner, Elchanan Solomon, and Paul Bendich. Improving metric dimensionality reduction with distributed topology. *arXiv preprint*, 2021. [arXiv:2106.07613](https://arxiv.org/abs/2106.07613).
- 39 Simon Zhang, Mengbai Xiao, Chengxin Guo, Liang Geng, Hao Wang, and Xiaodong Zhang. Hypha: A framework based on separation of parallelisms to accelerate persistent homology matrix reduction. In *Proceedings of the ACM International Conference on Supercomputing, ICS ’19*, pages 69–81, New York, NY, USA, 2019. Association for Computing Machinery. [doi:10.1145/3330345.3332147](https://doi.org/10.1145/3330345.3332147).

A Positive Fraction Erdős-Szekeres Theorem and Its Applications

Andrew Suk ✉

Department of Mathematics, University of California San Diego, La Jolla, CA, USA

Ji Zeng ✉

Department of Mathematics, University of California San Diego, La Jolla, CA, USA

Abstract

A famous theorem of Erdős and Szekeres states that any sequence of n distinct real numbers contains a monotone subsequence of length at least \sqrt{n} . Here, we prove a positive fraction version of this theorem. For $n > (k-1)^2$, any sequence A of n distinct real numbers contains a collection of subsets $A_1, \dots, A_k \subset A$, appearing sequentially, all of size $s = \Omega(n/k^2)$, such that every subsequence (a_1, \dots, a_k) , with $a_i \in A_i$, is increasing, or every such subsequence is decreasing. The subsequence $S = (A_1, \dots, A_k)$ described above is called *block-monotone of depth k and block-size s* . Our theorem is asymptotically best possible and follows from a more general Ramsey-type result for monotone paths, which we find of independent interest. We also show that for any positive integer k , any finite sequence of distinct real numbers can be partitioned into $O(k^2 \log k)$ block-monotone subsequences of depth at least k , upon deleting at most $(k-1)^2$ entries. We apply our results to mutually avoiding planar point sets and biarc diagrams in graph drawing.

2012 ACM Subject Classification Mathematics of computing → Combinatorics

Keywords and phrases Erdős-Szekeres, block-monotone, monotone biarc diagrams, mutually avoiding sets

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.62

Related Version *Full Version*: <https://arxiv.org/abs/2112.01750>

Funding *Andrew Suk*: Supported by NSF CAREER award DMS-1800746, NSF award DMS-1952786, and an Alfred Sloan Fellowship.

Ji Zeng: Supported by NSF grant DMS-1800746.

Acknowledgements The authors wish to thank the anonymous SoCG referees for their valuable suggestions.

1 Introduction

In 1935, Erdős and Szekeres [6] proved that any sequence of n distinct real numbers contains a monotone subsequence of length at least \sqrt{n} . This is a classical result in combinatorics and its generalizations and extensions have many important consequences in geometry, probability, and computer science. See Steele [13] for 7 different proofs along with several applications.

In this paper, we prove a positive fraction version of the Erdős-Szekeres theorem. We state this theorem using the following notion: A sequence $(a_1, a_2, \dots, a_{ks})$ of ks distinct real numbers is said to be *block-increasing* (*block-decreasing*) with *depth k* and *block-size s* if every subsequence $(a_{i_1}, a_{i_2}, \dots, a_{i_k})$, for $(j-1)s < i_j \leq js$, is increasing (decreasing). We call a sequence *block-monotone* if it's either block-increasing or block-decreasing.

► **Theorem 1.** *Let k and $n > (k-1)^2$ be positive integers. Then every sequence of n distinct real numbers contains a block-monotone subsequence of depth k and block-size $s = \Omega(n/k^2)$.*



© Andrew Suk and Ji Zeng;

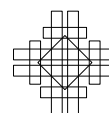
licensed under Creative Commons License CC-BY 4.0

38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 62; pp. 62:1–62:15

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



We prove Theorem 1 by establishing a more general Ramsey-type result for monotone paths, which we describe in detail in the next section. The theorem is also asymptotically best possible, see Remark 9.

By a repeated application of Theorem 1, we can decompose any sequence of n distinct real numbers into $O(k \log n)$ block-monotone subsequences of depth k upon deleting at most $(k-1)^2$ entries. Our next result shows that we can obtain such a partition, where the number of parts doesn't depend on n .

► **Theorem 2.** *For any positive integer k , every finite sequence of distinct real numbers can be partitioned into at most $O(k^2 \log k)$ block-monotone subsequences of depth at least k upon deleting at most $(k-1)^2$ entries.*

Our Theorem 2 is inspired by a similar problem of partitioning planar point sets into convex-positioned clusters, which is studied in [12]. A positive fraction Erdős-Szekeres-type result for convex polygons is given previously by Bárány and Valtr [3].

In the full version of this paper, we present a polynomial time algorithm that computes the block-monotone subsequence claimed by Theorem 1. Our proof of Theorem 2 is constructive hence implying a polynomial time algorithm for the claimed partition as well.

We give two applications of Theorems 1 and 2.

Mutually avoiding sets. Let A and B be finite point sets of \mathbb{R}^2 in *general position*, that is, no three points are collinear. We say that A and B are *mutually avoiding* if no line generated by a pair of points in A intersects the convex hull of B , and vice versa. Aronov et al. [1] used the Erdős-Szekeres Theorem to show that every n -element planar point set P in general position contains subsets $A, B \subset P$, each of size $\Omega(\sqrt{n})$, s.t. A and B are mutually avoiding. Valtr [14] showed that this bound is asymptotically best possible by slightly perturbing the points in an $\sqrt{n} \times \sqrt{n}$ grid. Following the same ideas of Aronov et al., we can use Theorem 1 to obtain the following.

► **Theorem 3.** *For every positive integer k there is a constant $\epsilon_k = \Omega(\frac{1}{k^2})$ s.t. every sufficiently large point set P in the plane in general position contains $2k$ disjoint subsets $A_1, \dots, A_k, B_1, \dots, B_k$, each of size at least $\epsilon_k |P|$, s.t. every pair of sets $A = \{a_1, \dots, a_k\}$ and $B = \{b_1, \dots, b_k\}$, with $a_i \in A_i$ and $b_i \in B_i$, are mutually avoiding.*

This improves an earlier result of Mirzaei and the first author [9], who proved the theorem above with $\epsilon_k = \Omega(\frac{1}{k^4})$. The result above is asymptotically best possible for both k and $|P|$: Consider a $k \times k$ grid G and replace each point with a cluster of $|P|/k^2$ points placed very close to each other so that the resulting point set P is in general position. If we can find subsets A_i 's and B_i 's as in Theorem 3, but each of size $\epsilon'_k |P|$ with $\epsilon'_k = \omega(\frac{1}{k^2})$, then we can find mutually avoiding subsets in G of size $\omega(k)$, contradicting Valtr's [14].

Finally, let us remark that a recent result due to Pach, Rubin, and Tardos [11] shows that every n -element planar point set in general position determines at least $n/e^{O(\sqrt{\log n})}$ pairwise crossing segments. By using Theorem 3 instead of Lemma 3.3 from their paper, one can improve the constant hidden in the O -notation.

Monotone biarc diagrams. A *proper arc diagram* is a drawing of a graph in the plane, whose vertices are points placed on the x -axis, called the *spine*, and each edge is drawn as a half-circle. A classic result of Bernhard and Kainen [4] shows that a planar graph admits a *planar proper arc diagram* if and only if it's a subgraph of a planar Hamiltonian graph. A *monotone biarc diagram* is a drawing of a graph in the plane, whose vertices are placed on a

spine, and each edge is drawn either as a half-circle or two half-circles centered on the spine, forming a continuous x -monotone biarc. See Figure 6 for an illustration. In [5], Di Giacomo et al. showed that every planar graph can be drawn as a *planar* monotone biarc diagram.

Using the Erdős-Szekeres Theorem, Bar-Yehuda and Fogel [2] showed that every graph $G = (V, E)$, with a given order on V , has a *double-paged book embedding* with at most $O(\sqrt{|E|})$ pages. That is, E can be partitioned into $O(\sqrt{|E|})$ parts, s.t. for each part E_i , (V, E_i) can be drawn as a planar monotone biarc diagram, and V appears on the spine with the given order. Our next result shows that we can significantly reduce the number of pages (parts), if we allow a small fraction of the pairs of edges to cross on each page.

► **Theorem 4.** *For any $\epsilon > 0$ and a graph $G = (V, E)$, where V is an ordered set, E can be partitioned into $O(\epsilon^{-2} \log(\epsilon^{-1}) \log(|E|))$ subsets E_i s.t. each (V, E_i) can be drawn as a monotone biarc diagram having no more than $\epsilon|E_i|^2$ crossing edge-pairs, and V appears on the spine with the given order.*

This paper is organized as follows: In Section 2, we prove Theorem 1 in the setting of monotone paths in multicolored ordered graphs. Section 3 is devoted to the proof of Theorem 2. In Section 4, we present proofs for the applications claimed above. Section 5 lists some remarks.

2 A positive fraction result for monotone paths

Several authors [7, 10, 8] observed that the Erdős-Szekeres theorem generalizes to the following graph-theoretic setting. Let G be a graph with vertex set $[n] = \{1, \dots, n\}$. A *monotone path of length k* in G is a k -tuple (v_1, \dots, v_k) of vertices s.t. $v_i < v_j$ for all $i < j$ and all edges $v_i v_{i+1}$, for $i \in [k-1]$, are in G .

► **Theorem 5.** *Let χ be a q -coloring of the pairs of $[n]$. Then there must be a monochromatic monotone path of length at least $n^{1/q}$.*

Given subsets $A, B \subset [n]$, we write $A < B$ if every element in A is less than every element in B .

► **Definition 6.** *Let G be a graph with vertex set $[n]$ and let $V_1, \dots, V_k \subset [n]$ and $p_1, \dots, p_{k+1} \in [n]$. Then we say that $(p_1, V_1, p_2, V_2, p_3, \dots, p_k, V_k, p_{k+1})$ is a *block-monotone path of depth k and block-size s* if*

1. $|V_i| = s$ for all i ,
2. we have $p_1 < V_1 < p_2 < V_2 < p_3 < \dots < p_k < V_k < p_{k+1}$,
3. and every $(2k+1)$ -tuple of the form

$$(p_1, v_1, p_2, v_2, \dots, p_k, v_k, p_{k+1}),$$

where $v_i \in V_i$, is a monotone path in G .

Our main result in this section is the following Ramsey-type theorem.

► **Theorem 7.** *There is an absolute constant $c > 0$ s.t. the following holds. Given integers $q \geq 2$, $k \geq 1$, and $n \geq (ck)^q$, let χ be a q -coloring of the pairs of $[n]$. Then χ produces a monochromatic block-monotone path of depth k and block-size $s \geq \frac{n}{(ck)^q}$.*

A careful calculation shows that we can take $c = 40$ in the theorem above. We will need the following lemma.

62:4 **A Positive Fraction Erdős-Szekeres Theorem and Its Applications**

► **Lemma 8.** *Let $q \geq 2$ and $N > 3^q$. Then for any q -coloring of the pairs of $[N]$, there is a monochromatic block-monotone path of depth 1 and block-size $s \geq \frac{N}{q3^{3q}}$.*

Proof. Let χ be a q -coloring of the pairs of $[N]$, and set $r = 3^q$. By Theorem 5, every subset of size r of $[N]$ gives rise to a monochromatic monotone path of length 3. Hence, χ produces at least

$$\frac{\binom{N}{r}}{\binom{N-3}{r-3}} \geq \frac{6}{r^3} \binom{N}{3}$$

monochromatic monotone paths of length 3 in $[N]$. Hence, there are at least $\frac{6}{qr^3} \binom{N}{3}$ monochromatic monotone paths of length 3, all of which have the same color. By averaging, there are two vertices $p_1, p_2 \in [N]$, s.t. at least $\frac{N}{qr^3}$ of these monochromatic monotone paths of length 3 start at vertex p_1 and ends at vertex p_2 . By setting V_1 to be the “middle” vertices of these paths, (p_1, V_1, p_2) is a monochromatic block-monotone path of depth 1 and block-size $s \geq \frac{N}{qr^3} = \frac{N}{q3^{3q}}$. ◀

Proof of Theorem 7. Let χ be a q -coloring of the pairs of $[n]$ and let c be a sufficiently large constant that will be determined later. Set $s = \lceil \frac{n}{(ck)^q} \rceil$. For the sake of contradiction, suppose χ does not produce a monochromatic block-monotone path of depth k and block-size s . For each element $v \in [n]$, we label v with $f(v) = (b_1, \dots, b_q)$, where b_i denotes the depth of the longest block-monotone path with block-size s in color i , ending at v . By our assumption, we have $0 \leq b_i \leq k - 1$, which implies that there are at most k^q distinct labels. By the pigeonhole principle, there is a subset $V \subset [n]$ of size at least n/k^q , s.t. the elements of V all have the same label.

By Lemma 8, there are vertices $p_1, p_2 \in V$, a subset $V' \subset V$, and a color α s.t. (p_1, V', p_2) is a monochromatic block-monotone path in color α , with block-size $t \geq \frac{|V|}{q3^{3q}}$. By setting c to be sufficiently large, we have

$$t \geq \frac{|V|}{q3^{3q}} \geq \frac{n}{k^q q 3^{3q}} \geq \left\lceil \frac{n}{(ck)^q} \right\rceil = s.$$

However, this contradicts the fact that $f(p_1) = f(p_2)$, since the longest supported monotone path with block-size s in color α ending at vertex p_1 can be extended to a longer one ending at p_2 . This completes the proof. ◀

Proof of Theorem 1. Let $A = (a_1, \dots, a_n)$ be a sequence of distinct real numbers. Let χ be a red/blue coloring of the pairs of A s.t. for $i < j$, we have $\chi(a_i, a_j) = \text{red}$ if $a_i < a_j$ and $\chi(a_i, a_j) = \text{blue}$ if $a_i > a_j$. In other words, we color increasing pairs by red and decreasing pairs by blue.

If $n < (ck)^2$, notice that $n/(ck)^2 < 1$. By our assumption $n > (k - 1)^2$, the classical Erdős-Szekeres theorem gives us a monotone subsequence in A of length at least k , which can be regarded as a block-monotone subsequence of depth at least k and block-size $s = 1 > n/(ck)^2$.

If $n \geq (ck)^2$, by Theorem 7, there is a monochromatic block-monotone path of depth k and block-size $s \geq n/(ck)^2$ in the complete graph on A , which can be regarded as a block-monotone subsequence of A with the claimed depth and block-size. ◀

► **Remark 9.** For each $k, q, s > 0$, the simple construction below shows Theorem 7 is tight up to the constant factor c^q . We first construct $K(k, q)$, for each k and q , a q -colored complete graph on $[k^q]$, whose longest monochromatic monotone path has length k : $K(k, 1)$ is just a monochromatic copy of the complete graph on $[k]$. To construct $K(k, q)$ from $K(k, q - 1)$,

take k copies of $K(k, q - 1)$ with the same set of $q - 1$ colors, place them in order and color the remaining edges by a new color. Now replace each point in $K(k, q)$ by a cluster of s points, where within each cluster one can arbitrarily color the edges. The resulting q -colored complete graph has no k subsets $V_1, V_2, \dots, V_k \subset [n]$ each of size $s + 1$ and edges between them monochromatic, otherwise $K(k, q)$ would have a monochromatic monotone path with length larger than k .

It's well-known that the sharpness of the classical Erdős-Szekeres theorem comes from sequences such as

$$S(k) = (k, k - 1, \dots, 1, 2k, 2k - 1, \dots, 2k + 1, \dots, k^2, k^2 - 1, k(k - 1) + 1).$$

We note that if we color the increasing pairs of $S(k)$ by red and the decreasing pairs of $S(k)$ by blue, we obtain the graph $K(k, 2)$. If we replace each entry $s_i \in S(k)$ by a cluster of s distinct real numbers very close to s_i , we obtain an example showing that Theorem 1 is asymptotically best possible.

3 Block-monotone sequence partition

This section is devoted to the proof of Theorem 2. We shall consider this problem geometrically by identifying each entry a_i of a given sequence $A = (a_i)_{i=1}^n$ as a planar point $(i, a_i) \in \mathbb{R}^2$. As we consider sequences of distinct real numbers, throughout this section, we assume that all point sets have the property that no two members share the same x -coordinate or the same y -coordinate.

Thus, we analogously define block-monotone point sets as follows: A set of ks planar points is said to be *block-increasing (block-decreasing)* with *depth* k and *block-size* s if it can be written as $\{(x_i, y_i)\}_{i=1}^{ks}$ s.t. $x_i < x_{i+1}$ for all i and every sequence $(y_{i_1}, y_{i_2}, \dots, y_{i_k})$, for $(j - 1)s < i_j \leq js$, is increasing (decreasing). We say that a point set is *block-monotone* if it's either block-increasing or block-decreasing. For each $j \in [k]$ we call the subset $\{(x_i, y_i)\}_{i=(j-1)s+1}^{js}$ the *j -th block* of this block-monotone point set.

Hence, Theorem 2 immediately follows from the following.

► **Theorem 10.** *For any positive integer k , every finite planar point set can be partitioned into at most $O(k^2 \log k)$ block-monotone point subsets of depth at least k and a remaining set of size at most $(k - 1)^2$.*

Given a point set $P \subset \mathbb{R}^2$, let

$$U(P) := \{(x, y) \in \mathbb{R}^2; y > y', \forall (x', y') \in P\}, \tag{up}$$

$$D(P) := \{(x, y) \in \mathbb{R}^2; y < y', \forall (x', y') \in P\}, \tag{down}$$

$$L(P) := \{(x, y) \in \mathbb{R}^2; x < x', \forall (x', y') \in P\}, \tag{left}$$

$$R(P) := \{(x, y) \in \mathbb{R}^2; x > x', \forall (x', y') \in P\}. \tag{right}$$

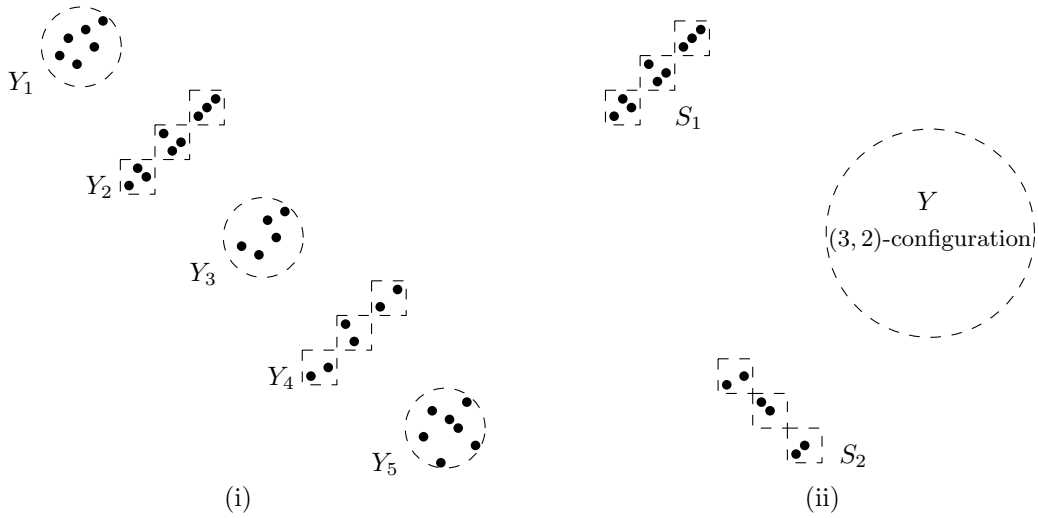
Our proof of Theorem 10 relies on the following definitions. The constant c below (and throughout this section) is from Theorem 7. See Figure 1 for an illustration.

► **Definition 11.** *A point set $P \subset \mathbb{R}^2$ is said to be a (k, t) -configuration if P can be written as a disjoint union of subsets $P = Y_1 \cup Y_2 \cup \dots \cup Y_{2t+1}$ s.t.*

- $\forall i \in [t], Y_{2i}$ is a block-monotone point set of depth k and block-size at least $|Y_{2j+1}| / (3ck)^2$ for all $j \in \{0\} \cup [t]$;
- either $\cup_{j=i+1}^{2t+1} Y_j$ is located entirely in $R(Y_i) \cap U(Y_i)$ for all $i \in [2t]$, or $\cup_{j=i+1}^{2t+1} Y_j$ is located entirely in $R(Y_i) \cap D(Y_i)$ for all $i \in [2t]$.

► **Definition 12.** A point set $P \subset \mathbb{R}^2$ is said to be a (k, l, t) -pattern if P can be written as a disjoint union of subsets $P = S_1 \cup S_2 \cup \dots \cup S_l \cup Y$ s.t.

- Y is a (k, t) -configuration;
- $\forall i \in [l], S_i$ is a block-monotone point set of depth k and block-size at least $|Y|/(3ck)^2$;
- $\forall i \in [l],$ the set $(\cup_{j=i+1}^l S_j) \cup Y$ is located entirely in one of the following regions: $U(S_i) \cap L(S_i), U(S_i) \cap R(S_i), D(S_i) \cap L(S_i)$ and $D(S_i) \cap R(S_i)$.



■ **Figure 1** (i) A $(3, 2)$ -configuration. (ii) A $(3, 2, 2)$ -pattern.

If a planar point set P is a $(k, 4k, t)$ -pattern or a (k, l, k) -pattern, the next two lemmas state that we can efficiently partition P into few block-monotone point sets of depth at least k and a small remaining set.

► **Lemma 13.** If P is a $(k, 4k, t)$ -pattern, then P can be partitioned into $O(k \log k)$ block-monotone point sets of depth at least k and a remaining set of size $O(k^2)$.

► **Lemma 14.** If P is a (k, l, k) -pattern, then P can be partitioned into $O(k^2 \log k + l)$ block-monotone point sets of depth at least k and a remaining set of size $O(k^3)$.

Starting with an arbitrary point set P , which can be regarded as a $(k, 0, 0)$ -pattern, we will repeatedly apply the following lemma until P is partitioned into few block-monotone point sets, a set P' that is either a $(k, 4k, t)$ -pattern or a (k, l, k) -pattern, and a small remaining set.

► **Lemma 15.** For $l < 4k$ and $t < k$, a (k, l, t) -pattern P can be partitioned into r block-monotone point sets with depth at least k , a point set P' , and a remaining set E s.t.

1. $r = O(k), |P'| \leq k(3k - 1)^2$ and $E = \emptyset$; or
2. $r = O(k \log k), P'$ is a $(k, l, t + 1)$ -pattern and $|E| = O(k^2)$; or
3. $r = O(k \log k), P'$ is a $(k, l + t, 0)$ -pattern and $|E| = O(k^2)$.

Moreover, when $t = 0$, we can always have this partition of P as in either case 1 or case 2.

Before we prove the lemmas above, let us use them to prove Theorem 10.

Proof of Theorem 10. Let P be the given point set. For $i \geq 0$, we inductively construct a partition $\mathcal{F}_i \cup \{P_i, E_i\}$ of P s.t.

- P_i is a (k, l_i, t_i) -pattern,
- $|E_i| = O(ik^2)$,
- \mathcal{F}_i is a disjoint family of block-monotone point sets of depth at least k , and $|\mathcal{F}_i| = O(ik \log k)$.

We start with $P_0 = P$, which is a $(k, 0, 0)$ -pattern, and $\mathcal{F}_0 = E_0 = \emptyset$. Suppose we have constructed the i -th partition $\mathcal{F}_i \cup \{P_i, E_i\}$ of P . If $|P_i| \leq k(3k - 1)^2$, or $l_i \geq 4k$, or $t_i \geq k$, we end this inductive construction process, otherwise, we construct the next partition $\mathcal{F}_{i+1} \cup \{P_{i+1}, E_{i+1}\}$ as follows.

According to Lemma 15, P_i can be partitioned into r block-monotone point sets with depth at least k , denoted as $\{P_{i,1}, \dots, P_{i,r}\}$, a point set P' , and a remaining set E , s.t. either one of the following cases happens.

Case 1. We have $r = O(k)$, $|P'| \leq k(3k - 1)^2$, and $E = \emptyset$. In this case, we define $\mathcal{F}_{i+1} = \mathcal{F}_i \cup \{P_{i,1}, \dots, P_{i,r}\}$, $P_{i+1} = P'$, and $E_{i+1} = E_i \cup E$. Notice that we have $|\mathcal{F}_{i+1}| = |\mathcal{F}_i| + O(k) = O((i + 1)k \log k)$ and $|E_{i+1}| = |E_i| + 0 = O((i + 1)k^2)$.

Case 2. We have $r = O(k \log k)$, P' is a $(k, l_i, t_i + 1)$ -pattern, and $|E| = O(k^2)$. In this case, we define $\mathcal{F}_{i+1} = \mathcal{F}_i \cup \{P_{i,1}, \dots, P_{i,r}\}$, $P_{i+1} = P'$, and $E_{i+1} = E_i \cup E$. This means $l_{i+1} = l_i$ and $t_{i+1} = t_i + 1$. Notice that we have $|\mathcal{F}_{i+1}| = |\mathcal{F}_i| + O(k \log k) = O((i + 1)k \log k)$ and $|E_{i+1}| = |E_i| + O(k^2) = O((i + 1)k^2)$.

Case 3. We have $r = O(k \log k)$, P' is a $(k, l_i + t_i, 0)$ -pattern, and $|E| = O(k^2)$. In this case, we define $\mathcal{F}_{i+1} = \mathcal{F}_i \cup \{P_{i,1}, \dots, P_{i,r}\}$, $P_{i+1} = P'$, and $E_{i+1} = E_i \cup E$. This means $l_{i+1} = l_i + t_i$ and $t_{i+1} = 0$. Again, we have $|\mathcal{F}_{i+1}| = O((i + 1)k \log k)$ and $|E_{i+1}| = O((i + 1)k^2)$.

When $t_i = 0$, by Lemma 15, we can always partition P_i as in Case 1 or Case 2. So we always construct $\mathcal{F}_{i+1} \cup \{P_{i+1}, E_{i+1}\}$ according to Case 1 or Case 2 when $t_i = 0$.

Let $\mathcal{F}_w \cup \{P_w, E_w\}$ be the last partition of P constructed in this process. Here, P_w is a (k, l_w, t_w) -pattern. We must have either $|P_w| \leq k(3k - 1)^2$, or $l_w \geq 4k$, or $t_w \geq k$. Since $t_{i+1} \leq t_i + 1$ and $l_{i+1} \leq l_i + t_i$ for all i , we have $t_w \leq k$ and $l_w \leq 5k$. Since we always construct the $(i + 1)$ -th partition according to Case 1 or Case 2 when $t_i = 0$, the sum $l_i + t_i$ always increases by at least 1 after 2 inductive process. So we have $w/2 \leq t_w + l_w \leq 6k$ and hence $w \leq 12k$.

Now we handle $\mathcal{F}_w \cup \{P_w, E_w\}$ based on how the construction process ends.

If the construction process ended with $|P_w| \leq k(3k - 1)^2$, we define $E_{w+1} = E_w \cup P_w$ and $\mathcal{F}_{w+1} = \mathcal{F}_w$. Since $w \leq 12k$, we have $|\mathcal{F}_{w+1}| = O(k^2 \log(k))$ and $|E_{w+1}| = O(k^3)$.

If the construction process ended with $l_w \geq 4k$, by Definition 12, we can partition P_w into $l_w - 4k$ many block-monotone point sets of depth k , denoted as $\{P_{w,1}, \dots, P_{w,l_w-4k}\}$, and a $(k, 4k, t_w)$ -pattern P'_w . Then, by Lemma 13, P'_w can be partitioned into $r = O(k \log k)$ block-monotone point sets of depth at least k , denoted as $\{P'_{w,1}, \dots, P'_{w,r}\}$, and a remaining set E of size $O(k^2)$. We define $E_{w+1} = E_w \cup E$ and

$$\mathcal{F}_{w+1} = \mathcal{F}_w \cup \{P_{w,1}, \dots, P_{w,l_w-4k}, P'_{w,1}, \dots, P'_{w,r}\}.$$

Using $w \leq 12k$ and other bounds we mentioned above, we can check $|\mathcal{F}_{w+1}| = O(k^2 \log(k))$ and $|E_{w+1}| = O(k^3)$.

If the construction process ended with $t_w \geq k$, we actually have $t_w = k$ and $l_w < 4k$. By Lemma 14, we can partition P_w into $r = O(k^2 \log(k) + l_w)$ block-monotone point sets of depth at least k , denoted as $\{P_{w,1}, \dots, P_{w,r}\}$, and a remaining set E of size $O(k^3)$. We define $E_{w+1} = E_w \cup E$ and $\mathcal{F}_{w+1} = \mathcal{F}_w \cup \{P_{w,1}, \dots, P_{w,r}\}$. Again, we can check $|\mathcal{F}_{w+1}| = O(k^2 \log(k))$ and $|E_{w+1}| = O(k^3)$.

Overall, we can always obtain a partition $\mathcal{F}_{w+1} \cup \{E_{w+1}\}$ of P with $|\mathcal{F}_{w+1}| = O(k^2 \log(k))$ and $|E_{w+1}| = O(k^3)$. Using the classical Erdős-Szekeres theorem, we can always find a monotone sequence of length at least k in E_{w+1} when $|E_{w+1}| > (k-1)^2$. By a repeated application of this fact, we can partition E_{w+1} into $O(k^2)$ block-monotone point sets of depth k and block-size 1, and a remaining set E of size at most $(k-1)^2$. We define \mathcal{F} to be the union of \mathcal{F}_{w+1} and these block-monotone sequences. The partition $\mathcal{F} \cup \{E\}$ of P has the desired properties and concludes the proof. \blacktriangleleft

We now give proofs for Lemmas 13, 14, and 15. We need the following facts.

► **Fact 16.** *For any positive integer k , every point set P can be partitioned into $O(k \log(k))$ block-monotone point sets of depth k and a remaining set P' with $|P'| \leq \max\{|P|/k, (k-1)^2\}$.*

This fact can be established by repeatedly using Theorem 1 to pull out block-monotone point sets and applying the elementary inequality $(1-x^{-1})^{x \log(x)} \leq x^{-1}$ for any $x > 1$.

► **Fact 17.** *For any positive integers k and m , every block-monotone point set P with depth k and $|P| \geq m$ can be partitioned into a block-monotone point set of depth k , a subset of size exactly m , and a remaining set of size less than k .*

This fact can be established by taking out $\lceil m/k \rceil$ points from each block of P . Then we have taken out $k \cdot \lceil m/k \rceil = m + r$ points, where $0 \leq r < k$.

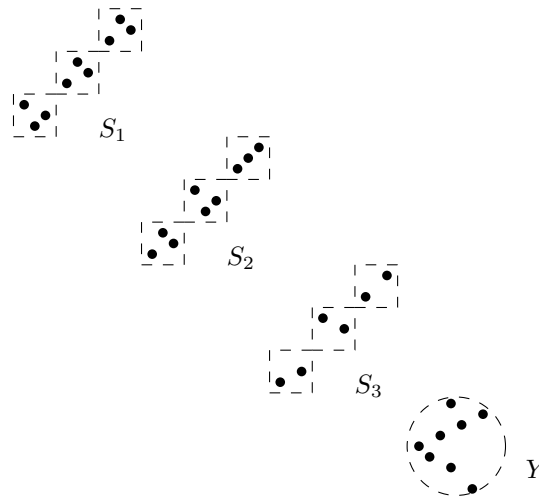
Proof of Lemma 13. Write the given $(k, 4k, t)$ -pattern $P = S_1 \cup \dots \cup S_{4k} \cup Y$ as in Definition 12. By definition, each block-monotone point set S_i is contained in one of the 4 regions: $U(Y) \cap L(Y)$, $U(Y) \cap R(Y)$, $D(Y) \cap L(Y)$ and $D(Y) \cap R(Y)$. By Pigeonhole principle, there are k indices i_1, \dots, i_k s.t. all S_{i_j} , for $j \in [k]$, are contained in one of the regions above. Without loss of generality, we assume S_1, \dots, S_k are all located entirely in $U(Y) \cap L(Y)$.

We have $S_{i_2} \subset D(S_{i_1}) \cap R(S_{i_1})$ for all $1 \leq i_1 < i_2 \leq k$. Indeed, since $Y \subset D(S_{i_1}) \cap R(S_{i_1})$, Definition 12 guarantees that $(\cup_{j=i_1+1}^k S_j) \cup Y$ to be contained in $D(S_{i_1}) \cap R(S_{i_1})$ and, in particular, S_{i_2} is contained in this region. See Figure 2 for an illustration.

Now apply Fact 16 to Y , we can partition Y into $\{A_1, \dots, A_w, Y'\}$, where $w = O(k \log(k))$, s.t. each A_j is block-monotone of depth $9c^2k$, and either $|Y'| \leq |Y|/(9c^2k)$ or $|Y'| \leq (9c^2k-1)^2$. If $|Y'| \leq (9c^2k-1)^2$, we have partitioned P into $O(k \log(k))$ block-monotone point sets of depth at least k , which are $\{A_1, \dots, A_w, S_1, \dots, S_{4k}\}$, and a remaining set Y' of size $O(k^2)$, as wanted.

If $|Y'| \leq |Y|/(9c^2k)$, by Definition 12 we have $|Y'| \leq |S_i|$ for $i \in [k]$. We can apply Fact 17 with $m := |Y'|$ to S_i to obtain a partition $S_i = S'_i \cup B_i \cup E_i$ where S'_i is block-monotone of depth k , $|B_i| = |Y'|$ and $|E_i| \leq k$. We observe that $C = B_1 \cup B_2 \cup \dots \cup B_k \cup Y'$ is block-monotone of depth $k+1$ by its construction. Then we have partitioned P into $O(k \log(k))$ many block-monotone point sets, which are $\{A_1, \dots, A_w, S'_1, \dots, S'_k, S_{k+1}, \dots, S_{4k}, C\}$, and a remaining set $E := \cup_{i=1}^k E_i$ of size $O(k^2)$, as wanted. \blacktriangleleft

Proof of Lemma 14. Write the given (k, l, k) -pattern $P = S_1 \cup \dots \cup S_l \cup Y$ as in Definition 12 and the (k, k) -configuration $Y = Y_1 \cup \dots \cup Y_{2k+1}$ as in Definition 11. Since each S_i is block-monotone of depth k , it suffices to partition Y into $O(k^2 \log(k))$ many block-monotone point sets of depth at least k and a remaining set of size $O(k^3)$.



■ **Figure 2** In proof of Lemma 13, $S_{i_2} \subset D(S_{i_1}) \cap R(S_{i_1})$ for $i_1 < i_2$.

For each $j \in \{0\} \cup [k]$, we apply Fact 16 to obtain a partition of Y_{2j+1} into $O(k \log(k))$ many block-monotone point sets of depth $9c^2k$ and a remaining set Y'_{2j+1} of size at most $|Y_{2j+1}|/(9c^2k)$ or at most $(9c^2k - 1)^2$. We can apply Fact 16 again to partition Y'_{2j+1} into $O(k \log(k))$ many block-monotone point sets of depth $k + 1$ and a remaining set Y''_{2j+1} with

$$|Y''_{2j+1}| \leq \max\{|Y_{2j+1}|/(9c^2k(k + 1)), (9c^2k - 1)^2\}. \tag{1}$$

Denote the block-monotone point sets produced in this process as $\{A_{j,x}; x \in [w_j]\}$, where $w_j = O(k \log(k))$.

Next we denote $J_1 := \{j \in \{0\} \cup [k]; |Y''_{2j+1}| > (9c^2k - 1)^2\}$ and $J_2 := (\{0\} \cup [k]) \setminus J_1$. For each $j \in J_1$ and $i \in [k]$, we must have

$$|Y''_{2j+1}| \leq |Y_{2j+1}|/(9c^2k(k + 1)) \leq |Y_{2i}|/(k + 1),$$

where the second inequality is by Definition 11. Hence $|Y_{2i}| \geq |\cup_{j \in J_1} Y''_{2j+1}|$. We can apply Fact 17 with $m := |\cup_{j \in J_1} Y''_{2j+1}|$ to Y_{2i} to obtain a partition $Y_{2i} = Y'_{2i} \cup B_i \cup E_i$ where Y'_{2i} is block-monotone of depth k , $|B_i| = m$, and $|E_i| \leq k$. Since $|B_i| = |\cup_{j \in J_1} Y''_{2j+1}|$, we can take a further partition $B_i = \cup_{j \in J_1} B_{j,i}$ with $|B_{j,i}| = |Y''_{2j+1}|$ for each $j \in J_1$. Then we observe that $C_j = B_{j,1} \cup \dots \cup B_{j,j} \cup Y''_{2j+1} \cup B_{j,j+1} \cup \dots \cup B_{j,k}$ is block-monotone of depth $k + 1$ for each $j \in J_1$ by its construction.

Finally, let $E := (\cup_{i=1}^k E_i) \cup (\cup_{j \in J_2} Y''_{2j+1})$, it easy to check that $E = O(k^3)$. So we have partitioned Y into $O(k^2 \log(k))$ many block-monotone point sets, which are

$$\{A_{j,x}\}_{j \in \{0\} \cup [k], x \in [w_j]} \cup \{C_j\}_{j \in J_1} \cup \{Y'_{2i}\}_{i \in [k]},$$

and a remaining set E of size $O(k^3)$, as wanted. ◀

Proof of Lemma 15. Write the given (k, l, t) -pattern $P = S_1 \cup \dots \cup S_l \cup Y$ as in Definition 12 and the (k, t) -configuration $Y = Y_1 \cup \dots \cup Y_{2t+1}$ as in Definition 11. Without loss of generality, we assume $\cup_{j=i+1}^{2t+1} Y_j$ is located entirely in $R(Y_i) \cap U(Y_i)$ for all $i \in [2t]$. We also assume that Y_1 has the largest size among $\{Y_{2j+1}; j \in \{0\} \cup [t]\}$ because other scenarios can be proved similarly.

If $|Y_1| \leq (3k-1)^2$, we can partition P into $r = l + t = O(k)$ many block-monotone point sets of depth k , which are $\{S_1, \dots, S_l, Y_2, Y_4, \dots, Y_{2t}\}$, and a remaining set $P' := \cup_{j=0}^t Y_{2j+1}$ of size at most $k(3k-1)^2$, since $t < k$. So we conclude the lemma in case (1).

Now we assume $|Y_1| > (3k-1)^2$. Apply Theorem 1 to extract a block-monotone point set $S \subset Y_1$ of depth $3k$ and block-size at least $|Y_1|/(3ck)^2$ and name the i -th block of S as B_i for $i \in [3k]$. Our proof splits into two cases: S being block-increasing or S being block-decreasing.

Case 1. Suppose S is block-increasing, write $S_{l+i} := Y_{2(t+1-i)}$ for each $i \in [t]$ and set $P' = S_1 \cup \dots \cup S_{l+t} \cup (Y_1 \setminus S)$. We can check that P' is a $(k, k+l, 0)$ -pattern by Definition 12. Let $Z := \cup_{j=1}^t Y_{2j+1}$. By an argument similar to (1), we can apply Fact 16 three times to partition Z into $\{A_1, \dots, A_w, Z'\}$, where $w = O(k \log(k))$, s.t. each A_i is block-monotone of depth at least k and $|Z'| \leq \max\{|Z|/(9c^2k^3), (9c^2k-1)^2\}$.

If $|Z'| \leq (9c^2k-1)^2$, let $E = Z'$. We have partitioned P into $O(k \log(k))$ block-monotone point sets of depth at least k , which are $\{A_1, \dots, A_w, S\}$, a $(k, k+l, 0)$ -pattern P' , and a remaining set E of size $O(k^2)$. So we conclude the lemma in case (3).

If $|Z'| \leq |Z|/(9c^2k^3)$, notice that $|Z| \leq k|Y_1|$ since $t < k$, we have $|Z'| \leq |Y_1|/(3ck)^2 \leq |B_i|$, for each $i \in [3k]$. We can take a partition $B_i = B'_i \cup B''_i$ with $|B'_i| = |Z'|$. We observe that $C := B'_1 \cup \dots \cup B'_{3k} \cup Z'$ is block-increasing of depth $3k+1$ and $S' := B''_1 \cup \dots \cup B''_{3k}$ is block-increasing of depth $3k$ by their construction. We have partitioned P into $O(k \log(k))$ block-monotone point sets of depth at least k , which are $\{A_1, \dots, A_w, C, S'\}$, and a $(k, k+l, 0)$ -pattern P' . So we conclude the lemma in case (3).

Case 2. Suppose S is block-decreasing, we choose two points in the following regions:

$$\begin{aligned} (x_1, y_1) &\in R(B_k) \cap D(B_k) \cap L(B_{k+1}) \cap U(B_{k+1}), \\ (x_2, y_2) &\in R(B_{2k}) \cap D(B_{2k}) \cap L(B_{2k+1}) \cap U(B_{2k+1}). \end{aligned}$$

Also we require x_1 or x_2 isn't the x -coordinate of any element in P , and y_1 or y_2 isn't the y -coordinate of any element in P . We use the lines $x = x_i$ and $y = y_i$ for $i = 1, 2$ to divide the plane into a 3×3 grid and label the regions $R_i, i = 1, \dots, 9$ as in Figure 3.

Let $C := B_{k+1} \cup \dots \cup B_{2k}$ and notice that C is block-monotone of depth k and block-size at least $|Y_1|/(3ck)^2$. Define

$$Y' := (R_7 \cap Y_1) \cup C \cup (R_3 \cap Y_1) \cup Y_2 \cup Y_3 \cup \dots \cup Y_{2t+1}.$$

We can check that Y' is a $(k, t+1)$ -configuration and $P' := S_1 \cup \dots \cup S_l \cup Y'$ is a $(k, l, t+1)$ -pattern according to Definitions 11 and 12.

Next, we set $Z_1 := (Y_1 \setminus S) \cap (R_5 \cup R_6 \cup R_8 \cup R_9)$ and $Z_2 := (Y_1 \setminus S) \cap (R_1 \cup R_2 \cup R_4)$. By an argument similar to (1), we can apply Fact 16 twice to partition Z_j into $\{A_{j,1}, \dots, A_{j,w_j}, Z'_j\}$, where $w_j = O(k \log(k))$, s.t. each $A_{j,x}$ is block-monotone of depth at least k and $|Z'_j| \leq \max\{|Z_j|/(3ck)^2, (9c^2k-1)^2\}$.

Writing $C_1 := B_1 \cup \dots \cup B_k$ and $C_2 = B_{2k+1} \cup \dots \cup B_{3k}$, then, for $j = 1, 2$, either $|Z'_j| = O(k^2)$ or $C_j \cup Z'_j$ can be partitioned into two block-decreasing point sets of depth at least k . Indeed, if $|Z'_1| > (9c^2k-1)^2$, we must have

$$|Z'_1| \leq |Z_1|/(3ck)^2 \leq |Y_1|/(3ck)^2 \leq |B_i|,$$

for each $i \in [k]$. Take a partition $B_i = B'_i \cup B''_i$ with $|B'_i| = |Z'_1|$, then we can observe $C_1 := B'_1 \cup \dots \cup B'_k \cup Z'_1$ is block-decreasing of depth $k+1$ and $C'_1 = B''_1 \cup \dots \cup B''_k$ is block-decreasing of depth k by their construction, as wanted. A similar argument applies to $C_2 \cup Z'_2$.

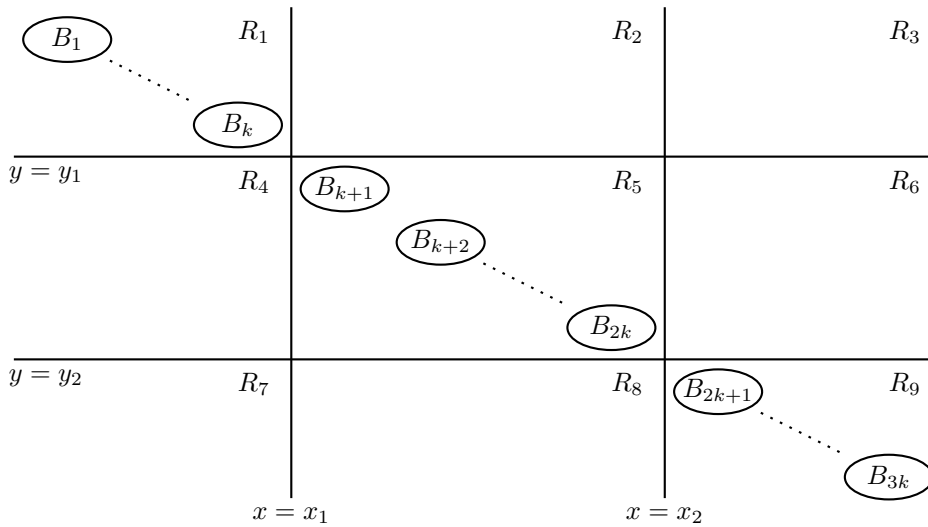


Figure 3 Division of the plane into 9 regions according to $(x_i, y_i), i = 1, 2$. Each ellipse represents a cluster of points as defined in the proof.

We have partitioned $P \setminus (C_1 \cup Z'_1 \cup C_2 \cup Z'_2)$ into $O(k \log(k))$ block-monotone sequence of depth at least k , which are $\{A_{j,x}; j = 1, 2, x \in [w_j]\}$, and a $(k, l, t + 1)$ -pattern P' . Combined with the claim in previous paragraph, we conclude the lemma in case (2).

Finally, when we are in the special case $t = 0$ and S is block-increasing, we can still use the arguments for the case when S is block-decreasing and conclude the lemma in case (2). The condition $t = 0$ can be used to verify Y' is a $(k, t + 1)$ -configuration, which is generally not true when $t > 0$ and S is block-increasing. ◀

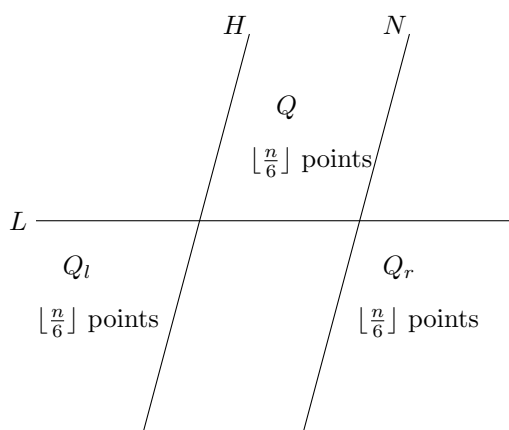
4 Applications

4.1 Mutually avoiding sets

We devote this subsection to the proof of Theorem 3. The proof is essentially the same as in [1], but we include it here for completeness. Given a non-vertical line L in the plane, we denote L^+ to be the closed upper-half plane defined by L , and L^- to be the closed lower-half plane defined by L . We need the following result, which is Lemma 1 in [1].

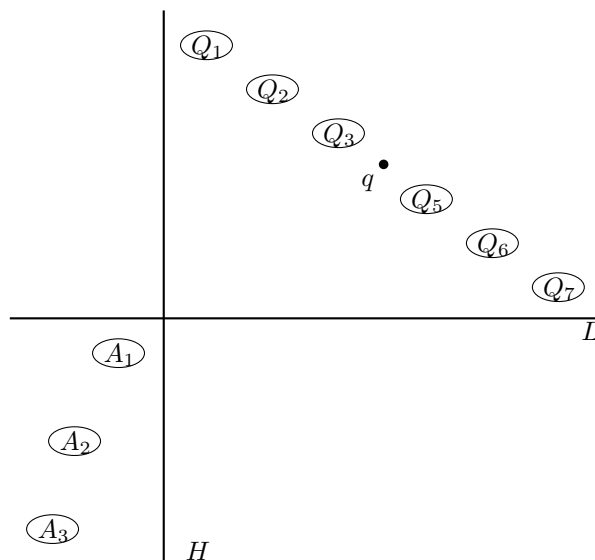
► **Lemma 18.** *Let $P, Q \subset \mathbb{R}^2$ be two n -element point sets with P and Q separated by a non-vertical line L and $P \cup Q$ in general position. Then for any positive integer $m \leq n$, there is another non-vertical line H s.t. $|H^+ \cap P| = |H^+ \cap Q| = m$ or $|H^- \cap P| = |H^- \cap Q| = m$.*

Proof of Theorem 3. Let k be as given and $n > 24k^2$. Let P be an n -element point set in the plane in general position. We start by taking a non-vertical line L to partition the plane s.t. each half-plane contains $\lfloor \frac{n}{2} \rfloor$ points from P . Then by Lemma 18, we obtain a non-vertical line H with, say, $H^+ \cap (L^+ \cap P) = H^+ \cap (L^- \cap P) = \lfloor \frac{n}{6} \rfloor$. Next, we find a third line N , by first setting $N = H$, and then sweeping N towards the direction of H^- , keeping it parallel with H , until $H^- \cap N^+ \cap L^+$ or $H^- \cap N^+ \cap L^-$ contains $\lfloor \frac{n}{6} \rfloor$ points from P . Without loss of generality, let us assume $Q := P \cap (H^- \cap N^+ \cap L^+)$ first reaches $\lfloor \frac{n}{6} \rfloor$ points, and the region $H^- \cap N^+ \cap L^-$ has less than $\lfloor \frac{n}{6} \rfloor$ points from P . Hence, both $Q_l := P \cap (H^+ \cap L^-)$ and $Q_r := P \cap (N^- \cap L^-)$ have at least $\lfloor \frac{n}{6} \rfloor$ points. See Figure 4 for an illustration.



■ **Figure 4** The division of plane into regions according to L, H, N .

We can apply an affine transformation so that L and H are perpendicular, and N is on the right side of H . Think of L as the x -axis, H as the y -axis, and N as a vertical line with a positive x -coordinate. After ordering the elements in Q according to their x -coordinates, we apply Theorem 1 to Q to obtain disjoint subsets $Q_1, \dots, Q_{2k+1} \subset Q$ s.t. (Q_1, \dots, Q_{2k+1}) is block-monotone of depth $2k + 1$ and block-size $\Omega(n/k^2)$, where each entry represents its y -coordinate. Without loss of generality, we can assume it is block-decreasing, otherwise we can work with Q_r rather than Q_l in the following arguments.



■ **Figure 5** An example when A_i 's are increasing. Each ellipse represents a cluster of points as defined in the proof.

Now fix a point $q \in Q_{k+1}$. We express the points in Q_l in polar coordinates (ρ, θ) with q being the origin. We can assume no two points in Q_l are at the same distance to q , otherwise a slight perturbation may be applied. By ordering the points in Q_l with respect to θ , in counter-clockwise order, we apply Theorem 1 to Q_l to obtain disjoint subsets $A_1, \dots, A_k \subset Q_l$ s.t. (A_1, \dots, A_k) is block-monotone of depth k and block-size $\Omega(n/k^2)$, where each entry

represents its distance to q . If it's block-decreasing, take $B_i = Q_i$ for $i \in [k]$, and if it's block-increasing, take $B_i = Q_{k+1+i}$. It is easy to check that the sets $\{A_1, \dots, A_k\}$ and $\{B_1, \dots, B_k\}$ have the claimed properties. See Figure 5 for an illustration. ◀

4.2 Monotone biarc diagrams

We devote this subsection to the proof of Theorem 4. Our proof is constructive, hence implying an recursive algorithm for the claimed outcome.

We start by making the simple observation that our main results hold for sequences of (not necessarily distinct) real numbers, if the term *block-monotone* now refers to being block-nondecreasing or block-nonincreasing. More precisely, a sequence $(a_1, a_2, \dots, a_{ks})$ of real numbers is said to be *block-nondecreasing* (*block-nonincreasing*) with *depth* k and *block-size* s if every subsequence $(a_{i_1}, a_{i_2}, \dots, a_{i_k})$, for $(j-1)s < i_j \leq js$, is nondecreasing (nonincreasing).

► **Theorem 19.** *For any positive integer k , every finite sequence of real numbers can be partitioned into at most $C_k = O(k^2 \log k)$ block-monotone subsequences of depth at least k upon deleting at most $(k-1)^2$ entries.*

To see our main results imply the above variation, it suffices to slightly perturb the possibly equal entries of a given sequence until all entries are distinct. Algorithms for our main results can also be applied after such a perturbation.

We need the following lemma in [2] for Theorem 4.

► **Lemma 20.** *For any graph $G = (V, E)$ with $V = [n]$, there exists $b \in [n]$ s.t. both the induced subgraphs of G on $\{1, 2, \dots, b\}$ and $\{b+1, b+2, \dots, n\}$ have no more than $|E|/2$ edges.*

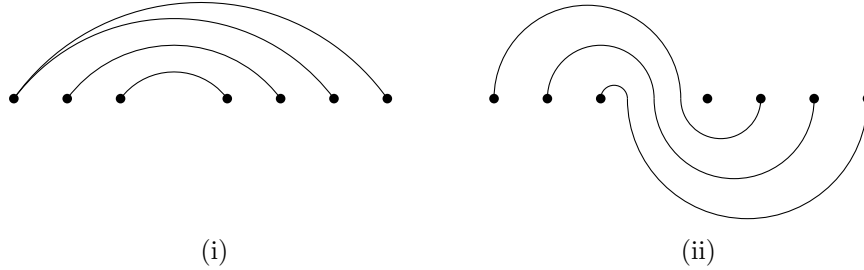
Proof. For $U \subset [n]$, let G_U denote the induced subgraph of G on U . Let b be the largest among $[n]$ s.t. $E(G_{[b]}) \leq \frac{|E|}{2}$, so $E(G_{[b+1]}) > \frac{|E|}{2}$. Notice that $E(G_{[b+1]})$ and $E(G_{[n] \setminus [b]})$ are two disjoint subsets of E , so $E(G_{[n] \setminus [b]}) \leq |E| - E(G_{[b+1]}) < \frac{|E|}{2}$, as wanted. ◀

Proof of Theorem 4. We prove by induction on $|E|$. The base case when $|E| = 1$ is trivial. For the inductive step, by the given order on V , we can identify V with $[n]$. We find such a b according to Lemma 20. Consider the set E' of edges between $[b]$ and $[n] \setminus [b]$. By writing each edge $e \in E'$ as (x, y) , where $x \in [b]$ and $y \in [n] \setminus [b]$, we order the elements in E' lexicographically: for $(x, y), (x', y') \in E'$, we have $(x, y) < (x', y')$ when $x < x'$ or when $x = x'$ and $y < y'$.

Given the order on E' described above, consider the sequence of right-endpoints in E' . We apply Theorem 19 with parameter $k = \lceil \epsilon^{-1} \rceil$ to this sequence, to decompose it into C_k many block-monotone sequences of depth k , upon deleting at most $(k-1)^2$ entries. For each block-monotone subsequence of depth k , we draw the corresponding edges on a single page as follows. If the subsequence is block-nonincreasing of depth k and block-size s , we draw the corresponding edges as semicircles above the spine. Then, two edges cross only if they come from the same block. Since there are a total of $\binom{ks}{2}$ pairs of edges, and only $k \binom{s}{2}$ such pairs from the same block, the fraction of pairs of edges that cross in such a drawing is at most $1/k$. See Figure 6(i). Similarly, if the subsequence is block-nondecreasing of depth k and block-size s , we draw the corresponding edges as monotone biarcs, consisting of two semicircles with the first (left) one above the spine, and the second (right) one below the spine. Furthermore,

we draw the monotone biarc s.t. it crosses the spine at $b + 1 - \ell/n - r/(2n^2)$, where ℓ and r are the left and right endpoints of the edge respectively. See Figure 6(ii). By the same argument above, the fraction of pairs of edges that cross in such a drawing is at most $1/k$.

Hence, E' can be decomposed into $C_k + (k - 1)^2$ many monotone biarc diagrams, s.t. each monotone biarc diagram has at most $1/k$ -fraction of pairs of edges that are crossing.



■ **Figure 6** (i) A proper arc diagram. (ii) A monotone biarc diagram.

For edges within $[b]$, Lemma 20 and the inductive hypothesis tell us that they can be decomposed into $(C_k + (k - 1)^2)(\log |E| - 1)$ monotone biarc diagrams, s.t. the fraction of pairs of edges that are crossing in each diagram is at most $1/k$. The same argument applies to the edges within $[n] \setminus [b]$. However, notice that two such monotone biarc diagrams, one in $[b]$ and another in $[n] \setminus [b]$, can be drawn on the same page without introducing more crossings. Hence, we can decompose $E \setminus E'$ into at most $(C_k + (k - 1)^2)(\log |E| - 1)$ such monotone biarc diagrams, giving us a total of $(C_k + (k - 1)^2) \log |E|$ monotone biarc diagrams. ◀

5 Final remarks

1. We call a sequence (a_1, a_2, \dots, a_n) of n distinct real numbers ϵ -increasing (ϵ -decreasing) if the number of decreasing (increasing) pairs (a_i, a_j) , where $i < j$, is less than ϵn^2 . And we call a sequence ϵ -monotone if it's either ϵ -increasing or ϵ -decreasing. Clearly, a block-monotone sequence of depth k is an ϵ -monotone sequence with $\epsilon = k^{-1}$. Hence, Theorem 1 implies the following.

► **Corollary 21.** *For all $n > 0$ and $\epsilon > 0$, every sequence of n distinct real numbers contains an ϵ -monotone subsequence of length at least $\Omega(\epsilon n)$.*

This corollary is also asymptotically best possible. To see this, for $n > (k - 1)^2$ and a sequence $A = (a_i)_{i=1}^n$ of distinct real numbers, we can apply Corollary 21 with $\epsilon = (64k)^{-1}$ to A and obtain an ϵ -monotone subsequence $S \subset A$ and then apply Lemma 2.1 in [11] to S to obtain a block-monotone subsequence of depth k and block-size $\Omega(n/k^2)$. So Corollary 21 implies Theorem 1.

2. Let $f(k)$ be the smallest number N s.t. every finite sequence of distinct real numbers can be partitioned into at most N block-monotone subsequences of depth at least k upon deleting $(k - 1)^2$ entries. Our Theorem 2 is equivalent to saying $f(k) = O(k^2 \log(k))$. The $K(k, 2)$ -type construction in Remark 9 implies $f(k) \geq k$. What is the asymptotic order of $f(k)$?

3. We suspect our algorithm for Theorem 1 presented in the full version of this paper can be improved. How fast can we compute a block-monotone subsequence as large as claimed in Theorem 1? Can we do it within time almost linear in n for all k ?

References

- 1 Boris Aronov, Paul Erdős, Wayne Goddard, Daniel J. Kleitman, Michael Klugerman, János Pach, and Leonard J. Schulman. Crossing families. In *Proceedings of the seventh annual symposium on Computational geometry*, pages 351–356, 1991.
- 2 Reuven Bar-Yehuda and Sergio Fogel. Partitioning a sequence into few monotone subsequences. *Acta Informatica*, 35(5):421–440, 1998.
- 3 Imre Bárány and Pavel Valtr. A positive fraction Erdős-Szekeres theorem. *Discrete & Computational Geometry*, 19(3):335–342, 1998.
- 4 Frank Bernhart and Paul C. Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27(3):320–331, 1979.
- 5 Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Stephen K. Wismath. Curve-constrained drawings of planar graphs. *Computational Geometry*, 30(1):1–23, 2005.
- 6 P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.
- 7 Jacob Fox, János Pach, Benny Sudakov, and Andrew Suk. Erdős-Szekeres-type theorems for monotone paths and convex bodies. *Proceedings of the London Mathematical Society*, 105(5):953–982, May 2012.
- 8 Kevin Milans, Derick Stolee, and Douglas West. Ordered Ramsey theory and track representations of graphs. *Journal of Combinatorics*, 6(4):445–456, 2015.
- 9 Mozghan Mirzaei and Andrew Suk. A positive fraction mutually avoiding sets theorem. *Discrete Mathematics*, 343(3):111730, 2020.
- 10 Guy Moshkovitz and Asaf Shapira. Ramsey Theory, integer partitions and a new proof of the Erdős-Szekeres Theorem. *Advances in Mathematics*, 262:1107–1129, 2014.
- 11 János Pach, Natan Rubin, and Gábor Tardos. Planar point sets determine many pairwise crossing segments. *Advances in Mathematics*, 386:107779, 2021.
- 12 Attila Pór and Pavel Valtr. The partitioned version of the Erdős-Szekeres theorem. *Discrete & Computational Geometry*, 28(4):625–637, 2002.
- 13 J. Michael Steele. Variations on the monotone subsequence theme of Erdős and Szekeres. In *Discrete Probability and Algorithms*, pages 111–131, New York, NY, 1995. Springer New York.
- 14 Pavel Valtr. On mutually avoiding sets. In *The Mathematics of Paul Erdős II*, pages 324–328. Springer, 1997.

Optimal Coreset for Gaussian Kernel Density Estimation

Wai Ming Tai ✉

University of Chicago, IL, USA

Abstract

Given a point set $P \subset \mathbb{R}^d$, the kernel density estimate of P is defined as

$$\bar{\mathcal{G}}_P(x) = \frac{1}{|P|} \sum_{p \in P} e^{-\|x-p\|^2}$$

for any $x \in \mathbb{R}^d$. We study how to construct a small subset Q of P such that the kernel density estimate of P is approximated by the kernel density estimate of Q . This subset Q is called a coreset. The main technique in this work is constructing a ± 1 coloring on the point set P by discrepancy theory and we leverage Banaszczyk's Theorem. When $d > 1$ is a constant, our construction gives a coreset of size $O\left(\frac{1}{\epsilon}\right)$ as opposed to the best-known result of $O\left(\frac{1}{\epsilon} \sqrt{\log \frac{1}{\epsilon}}\right)$. It is the first result to give a breakthrough on the barrier of $\sqrt{\log}$ factor even when $d = 2$.

2012 ACM Subject Classification Theory of computation \rightarrow Design and analysis of algorithms

Keywords and phrases Discrepancy Theory, Kernel Density Estimation, Coreset

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.63

Related Version *Full Version:* <https://arxiv.org/abs/2007.08031>

1 Introduction

Kernel density estimation is a non-parametric way to estimate a probability distribution. Given a point set $P \subset \mathbb{R}^d$, the kernel density estimate (KDE) of P smooths out P to a continuous function [35, 36]. More precisely, given a point set $P \subset \mathbb{R}^d$ and a kernel $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, KDE is defined as the function $\bar{\mathcal{G}}_P(x) = \frac{1}{|P|} \sum_{p \in P} K(x, p)$ for any $x \in \mathbb{R}^d$. Here, the point x is called a *query*. One common example of kernel K is the Gaussian kernel, which is $K(x, y) = e^{-\|x-y\|^2}$ for any $x, y \in \mathbb{R}^d$, and it is the main focus of this paper. A wide range of application includes outlier detection [41], clustering [33], topological data analysis [32, 10], spatial anomaly detection [1, 18], statistical hypothesis test [17] and other [19, 23].

Generally speaking, the techniques using kernels are called *kernel methods*, in which KDE is the central role in these techniques. Kernel methods are prevalent in machine learning and statistics and often involve optimization problems. Optimization problems are generally hard in the sense that solving them usually has a super-linear or even an exponential dependence on the input's size in its running time. Therefore, reducing the size of the input will be desirable. A straightforward way to achieve this is extracting a small subset Q of the input P . This paper will study the construction of the subset Q such that $\bar{\mathcal{G}}_Q$ approximates $\bar{\mathcal{G}}_P$.

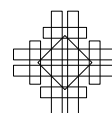
Classically, statisticians concern about different types of error such as L_1 -error [14] or L_2 -error [35, 36]. However, there are multiple modern applications that require L_∞ -error such as preserving classification margin [34], density estimation [40], topology [32] and hypothesis test on distributions [17]. For example, in topological data analysis, we might want to study the persistent homology of a super-level set of a kernel density estimate. In this case, L_∞ -error plays an important role here since a small perturbation could cause a significant change in its persistence diagram. Formally, we would like to solve the following problem.



© Wai Ming Tai;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 63; pp. 63:1–63:15



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Given a point set $P \subset \mathbb{R}^d$ and $\varepsilon > 0$, we construct a subset Q of P such that

$$\sup_{x \in \mathbb{R}^d} |\bar{\mathcal{G}}_P(x) - \bar{\mathcal{G}}_Q(x)| = \sup_{x \in \mathbb{R}^d} \left| \frac{1}{|P|} \sum_{p \in P} e^{-\|x-p\|^2} - \frac{1}{|Q|} \sum_{q \in Q} e^{-\|x-q\|^2} \right| \leq \varepsilon.$$

Then, how small can the size of Q , $|Q|$, be?

We call this subset Q an ε -coreset.

1.1 Known results

We now discuss some previous results for the size of an ε -coreset.

Josh et al. [20] showed that random sampling can achieve the size of $O(\frac{d}{\varepsilon^2})$. They investigated the VC-dimension of the super-level sets of a kernel and analyzed that the sample size can be bounded by it. In particular, the super-level sets of the Gaussian kernel are balls in \mathbb{R}^d . It reduces the problem to bounding the sample size of the range space of balls.

Lopaz-Paz et al. [24] later proved that the size of the coreset can be reduced to $O(\frac{1}{\varepsilon^2})$ by random sampling. They studied the reproducing kernel Hilbert space (RKHS) associated with a positive-definite kernel [3, 39, 38]. Note that the Gaussian kernel is a positive-definite kernel. In RKHS, one can bound the L_∞ -error between two KDEs of point sets P and Q by the kernel distance of P and Q . They showed that the sample size of $O(\frac{1}{\varepsilon^2})$ is sufficient to bound the kernel distance.

Other than random sampling, Lacoste-Julien et al. [22] showed a greedy approach can also achieve the size of $O(\frac{1}{\varepsilon^2})$. They applied Frank-Wolfe algorithm [13, 15] in RKHS to bound the error of the kernel distance.

Note that all of the above results have a factor of $\frac{1}{\varepsilon^2}$. Josh et al. [20] first showed that a sub- $O(\frac{1}{\varepsilon^2})$ result can be obtained by reducing the problem to constructing an ε -approximation for the range space of balls [26]. They assumed that d is constant. For the case of $d = 1$, their result gives the size of $O(\frac{1}{\varepsilon})$.

Later, Phillips [29] improved the result to $O((\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon})^{\frac{d}{d+2}})$ for constant d via geometric matching. It is based on the discrepancy approach. Namely, they construct a ± 1 coloring on the point set, recursively drop the points colored -1 and construct another ± 1 coloring on the points colored $+1$. We will discuss it in more detail below. Notably, for the case of $d = 2$, their bound is $O(\frac{1}{\varepsilon} \sqrt{\log \frac{1}{\varepsilon}})$ which is nearly-optimal (as a preview, the optimal bound is $\Omega(\frac{1}{\varepsilon})$) and is the first nearly-linear result for the case of $d > 1$.

Recently, Phillips and Tai [30] further improved the size of a coreset to $O(\frac{1}{\varepsilon} \log^d \frac{1}{\varepsilon})$ for constant d . It is also based on the discrepancy approach. They exploited the fact that the Gaussian kernel is multiplicatively separable. It implies that the Gaussian kernel can be rewritten as the weighted average of a family of axis-parallel boxes in \mathbb{R}^d . Finally, they reduced the problem to Tushnádý's problem [6, 2].

Also, Phillips and Tai [31] proved a nearly-optimal result of $O(\frac{\sqrt{d}}{\varepsilon} \sqrt{\log \frac{1}{\varepsilon}})$ shortly after that. It is also based on the discrepancy approach. They observed that the underlying structure of the positive-definite kernel allows us to bound the norm of the vectors and apply the lemma in [27], which used Banaszczyk's Theorem [4, 5]. Recall that the Gaussian kernel is a positive-definite kernel.

Except for the upper bound, there are some results on the lower bound for the size of an ε -coreset. Phillips [29] provided the first lower bound for the size of a coreset. They proved a lower bound of $\Omega(\frac{1}{\varepsilon})$ by giving an example that all points are spread out. When assuming

$d > \frac{1}{\varepsilon^2}$, Phillips and Tai [30] gave another example that forms a simplex and showed a lower bound of $\Omega(\frac{1}{\varepsilon^2})$. Later, Phillips and Tai [31] combined the techniques of the above two results and showed the lower bound of $\Omega(\frac{\sqrt{d}}{\varepsilon})$.

There are other conditional bounds for this problem. We suggest the readers refer to [31] for a more extensive review. Recently, Karnin and Liberty [21] defined the notion of Class Discrepancy which governs the coresets-complexity of different families of functions. Specifically, for analytic functions of squared distances (such as the Gaussian kernel), their analysis gives a discrepancy bound $D_m = O(\frac{\sqrt{d}}{m})$ which gives a coreset of size $O(\frac{\sqrt{d}}{\varepsilon})$. Their approach also used the discrepancy technique or, more precisely, Banaszczyk's Theorem [4, 5]. Unfortunately, their analysis requires *both* the point set P and the query x lie in a ball of a fixed radius R . Therefore, their result has a dependence on R . Strictly speaking, their result is not comparable to ours. It is not clear how to remove this assumption of R based on their result. Also, the lower bound constructions in [29, 31] rely on the fact that P is in an unbounded region and hence it is not clear how their result is comparable to the existing lower results.

1.2 Related works

In computational geometry, an ε -approximation is the approximation of a general set by a smaller subset. Given a set S and a collection \mathcal{C} of subsets of S , a subset $A \subset S$ is called an ε -approximation if $|\frac{|T|}{|S|} - \frac{|T \cap A|}{|A|}| \leq \varepsilon$ for all $T \in \mathcal{C}$. The pair (S, \mathcal{C}) is called a set system (also known as a range space or a hypergraph). One can rewrite the above guarantee as $|\frac{1}{|S|} \sum_{x \in S} \mathbb{1}_T(x) - \frac{1}{|A|} \sum_{x \in A} \mathbb{1}_T(x)| \leq \varepsilon$ where $\mathbb{1}_T$ is the indicator function of set T . If we replace this indicator function by a kernel such as the Gaussian kernel, it is the same as our ε -coreset. There is a rich history on the construction of an ε -approximation [11, 26]. One notable method is discrepancy theory, which is also our main technique. There is a wide range of techniques employed in this field. In the early 1980s, Beck devised the technique of partial coloring [7], and later a refinement of this technique called entropy method was introduced by Spencer [37]. The entropy method is first used to solve the famous ‘‘six standard deviations’’ theorem: given a set system of n points and n subsets, there is a coloring of discrepancy at most $6\sqrt{n}$. In contrast, random coloring gives the discrepancy of $O(\sqrt{n \log n})$. A more geometric example in discrepancy theory is Tuskányi's problem. It states that, given point set P of size n in \mathbb{R}^d , construct a ± 1 coloring σ on P such that the discrepancy $\min_{\sigma} \max_R |\sum_{P \cap R} \sigma(p)|$ is minimized where \max_R is over all axis-parallel boxes R . One previous approach of our ε -coreset problem reduces the problem to Tuskányi's problem.

On the topic of approximating KDE, Fast Gauss Transform [16] is a method to preprocess the input point set such that the computation of KDE at a query is faster than the brute-force approach. The idea in this method is expanding the Gaussian kernel by Hermite polynomials and truncating the expansion. Assuming that the data set lies inside a bounded region, the query time in this method is poly-logarithmic of n for constant dimension d . Also, Charikar et al. [9] studied the problem of designing a data structure that preprocesses the input to answer a KDE query in a faster time. They used locality-sensitive hashing to perform their data structure. However, the guarantee they obtained is a relative error, while ours is an additive error. More precisely, given a point set $P \subset \mathbb{R}^d$, Charikar et al. designed a data structure such that, for any query $x' \in \mathbb{R}^d$, the algorithm answers the value $\bar{G}_P(x') = \sum_{p \in P} e^{-\|x' - p\|^2}$ within $(1 + \varepsilon)$ -relative error. Also, the query time of their data structure is sublinear of n .

1.3 Our result

We construct an ε -coreset and bound the size of the ε -coreset via discrepancy theory. Roughly speaking, we construct a ± 1 coloring on our point set such that its discrepancy is small. Then, we drop the points colored -1 and recursively construct a ± 1 coloring on the points colored $+1$. Eventually, the remaining point set is the desired coreset. A famous theorem in discrepancy theory is Banaszczyk's Theorem [4, 5]. We will use Banaszczyk's Theorem to construct a coloring and prove the discrepancy is small by induction. To the best of our knowledge, this induction analysis combining with Banaszczyk's Theorem has not been seen in discrepancy theory before. In the constant dimensional space, we carefully study the structure of the Gaussian kernel and it allows us to construct an ε -coreset of size $O(1/\varepsilon)$. Our result is the first result to break the barrier of $\sqrt{\log}$ factor even when $d = 2$.

► **Theorem 1.** *Suppose $P \subset \mathbb{R}^d$ a point set of size n . Let $\bar{\mathcal{G}}_P$ be the Gaussian kernel density estimate of P , i.e. $\bar{\mathcal{G}}_P(x) = \frac{1}{|P|} \sum_{p \in P} e^{-\|x-p\|^2}$ for any $x \in \mathbb{R}^d$. For a fixed constant d , there is an algorithm that constructs a subset $Q \subset P$ of size $O(\frac{1}{\varepsilon})$ such that $\sup_{x \in \mathbb{R}^d} |\bar{\mathcal{G}}_P(x) - \bar{\mathcal{G}}_Q(x)| < \varepsilon$ and has a polynomial running time in n .*

Even if $d = 1$, the best known result is $O(1/\varepsilon)$ by [20, 30], which is optimal. Their approach is to reduce the problem to Tusnady's problem. A trivial solution of Tusnady's problem (and hence our problem) is: sort P and assign ± 1 on each point alternately. However, it is not clear that how this simple solution can be generalized to the higher dimensional case. Our algorithm gives a non-trivial perspective even though the optimal result was achieved previously.

2 Preliminaries

Our approach for constructing a coreset relies on discrepancy theory, which is a similar technique in range counting coreset [12, 28, 8]. We first introduce an equivalent problem (up to a constant factor) as follows.

Given a point set $P \subset \mathbb{R}^d$, what is the smallest quantity of $\sup_{x \in \mathbb{R}^d} |\sum_{p \in P} \sigma(p) e^{-\|x-p\|^2}|$ over all σ in the set of colorings from P to $\{-1, +1\}$?

Now, one can intuitively view the equivalence in the following way. If we rewrite the objective as:

$$\frac{1}{|P|} \left| \sum_{p \in P} \sigma(p) e^{-\|x-p\|^2} \right| = \left| \frac{1}{|P|} \sum_{p \in P} e^{-\|x-p\|^2} - \frac{1}{|P|/2} \sum_{p \in P_+} e^{-\|x-p\|^2} \right|$$

where $P_+ \subset P$ is the set of points that is assigned $+1$, then we can apply the halving technique [12, 28] which recursively invokes the coloring algorithm and retains the points assigned $+1$ until the subset of the desired size remains. Note that there is no guarantee that half of the points are assigned $+1$, while the other half is assigned -1 . However, we can handle this issue by some standard techniques [26] or see our proof for details.

Also, we define the following notations. Given a point set $P \subset \mathbb{R}^d$, a coloring $\sigma : P \rightarrow \{-1, +1\}$ and a point $x \in S$, we define the signed discrepancy $\mathcal{D}_{P,\sigma}(x)$ as

$$\mathcal{D}_{P,\sigma}(x) = \sum_{p \in P} \sigma(p) e^{-\|x-p\|^2}$$

It is worth noting that we expect $|\mathcal{D}_{P,\sigma}(x)| < O(1)$ in order to construct an ε -coreset of size $O(\frac{1}{\varepsilon})$ via this halving technique.

An important result in discrepancy theory is Banaszczyk's Theorem [4].

► **Theorem 2** (Banasczyk's Theorem [4]). *Suppose we are given a convex body $K \subset \mathbb{R}^m$ of the Gaussian measure at least $\frac{1}{2}$ and n vectors $v^{(1)}, v^{(2)}, \dots, v^{(n)} \in \mathbb{R}^m$ of norm at most 1, there is a coloring $\sigma : [n] \rightarrow \{-1, +1\}$ such that the vector $\sum_{i=1}^n \sigma(i)v^{(i)} \in cK = \{c \cdot y \mid y \in K\}$. Here, c is an absolute constant and the Gaussian measure of a convex body K is defined as $\int_{x \in K} \frac{1}{(2\pi)^{d/2}} e^{-\|x\|^2/2} dx$.*

The original proof of this theorem is non-constructive. Bansal et al. [5] proved that there is an efficient algorithm to construct the coloring in Banasczyk's Theorem. Moreover, assuming $m < n$, the running time is $O(n^{\omega+1})$ where ω is the exponent of matrix multiplication.

► **Theorem 3** (Constructive version of Banasczyk's Theorem [5]). *Suppose we are given n vectors $v^{(1)}, \dots, v^{(n)} \in \mathbb{R}^m$ of norm at most 1, there is an efficient randomized algorithm that constructs a coloring σ on P with the following guarantee: there are two absolute constants C', C'' such that, for any unit vector $\theta \in \mathbb{R}^m$ and $\alpha > 0$, we have*

$$\Pr[|\langle \theta, X \rangle| > \alpha] < C' e^{-C'' \alpha^2}$$

where X is the random variable of $\sum_{i=1}^n \sigma(i)v^{(i)}$. The probability in the above statement is distributed over all ± 1 colorings.

Finally, we introduce a useful theorem which is Markov Brother's Inequality.

► **Theorem 4** (Markov Brother's Inequality [25]). *Let $\mathcal{P}(x)$ be a polynomial of degree ρ . Then,*

$$\sup_{x \in [0,1]} |\mathcal{P}'(x)| \leq 2\rho^2 \sup_{x \in [0,1]} |\mathcal{P}(x)|$$

Here, \mathcal{P}' is the derivative of \mathcal{P} .

3 Proof overview

As we mentioned before, our equivalent problem statement suggests that we need to construct a ± 1 coloring on the input point set such that the absolute value of the signed discrepancy at all points is small. In this section, we will give an overview on how we construct the coloring and how it gives us the desired guarantees.

For exposition purposes, we illustrate the idea for the case of $d = 1$ even though previous results [20, 30] showed this case is trivial. Recall that our problem definition is: given a point set $P \subset \mathbb{R}$ of size n , construct a ± 1 coloring σ on P such that the absolute value of the signed discrepancy

$$|\mathcal{D}_{P,\sigma}(x)| = \left| \sum_{p \in P} \sigma(p) e^{-(x-p)^2} \right|$$

is bounded from above by a constant for all $x \in \mathbb{R}$.

Some general observations. We first make some observations. Note that $\mathcal{D}_{P,\sigma}$ is a smooth function of x that the slope at any x is bounded. It means that if $|\mathcal{D}_{P,\sigma}(x_0)|$ is small for some point x_0 then $|\mathcal{D}_{P,\sigma}(y)|$ is also small for any point y at a neighborhood of x_0 . Another observation is that $\mathcal{D}_{P,\sigma}$ is basically a linear combination of Gaussians and hence $|\mathcal{D}_{P,\sigma}(x)|$ is small for any x that is far away from all points in P .

Combining these two observations, if we lay down a grid on \mathbb{R} and consider the grid points that is not too far away from P , then we only need to construct a coloring σ such that $|\mathcal{D}_{P,\sigma}(x)|$ is small for all x in a *finite* set and it implies that $|\mathcal{D}_{P,\sigma}(x)|$ is small for all $x \in \mathbb{R}$. It is crucial because we preview that our algorithm for constructing the coloring σ is a randomized algorithm and the size of the finite set controls the number of events when we apply the union bound. Note that these observations hold for *any* coloring.

Techniques from the previous result. Now, we make the above observations more quantitative. Since the slope of each Gaussian at any point is bounded by $O(1)$ and there are n Gaussians in $\mathcal{D}_{P,\sigma}$, by triangle inequality, the absolute value of the slope of $\mathcal{D}_{P,\sigma}$ at any point is bounded by $O(n)$. Hence, if $|\mathcal{D}_{P,\sigma}(x_0)|$ is bounded by α for any point x_0 for any α then $|\mathcal{D}_{P,\sigma}(y)|$ is bounded by $\alpha + O(1)$ for all y that $|x_0 - y| < O(1/n)$. Also, Gaussians decay exponentially and hence $|\mathcal{D}_{P,\sigma}(x)| < O(1)$ for any x that $|x - p| > \Omega(\sqrt{\log n})$ for all $p \in P$.

If a coloring σ satisfies that

$$|\mathcal{D}_{P,\sigma}(x)| < \alpha \text{ for any } x \in \mathbb{R} \text{ with probability at least } 1 - O(e^{-\Omega(\alpha^2)}) \text{ for any } \alpha \quad (1)$$

then it implies, by union bound, this coloring σ satisfies that

$$|\mathcal{D}_{P,\sigma}(x)| < \alpha + O(1) \text{ for all } x \in \mathbb{R} \text{ with probability at least } 1 - N \cdot O(e^{-\Omega(\alpha^2)})$$

where N is the number of grid points that are in the grid of cell width $\Omega(1/n)$ and lie around some point in P within a radius of $O(\sqrt{\log n})$. The number N is bounded by $O(n^2\sqrt{\log n})$ because for each point $p \in P$ there are $O(\sqrt{\log n}/(1/n)) = O(n\sqrt{\log n})$ grid points around p within a radius of $O(\sqrt{\log n})$ and there are n points in P . By setting $\alpha = O(\sqrt{\log n})$, we have

$$|\mathcal{D}_{P,\sigma}(x)| < O(\sqrt{\log n}) \text{ for all } x \in \mathbb{R} \text{ with probability at least } 1 - 1/10$$

if we manage to construct a coloring σ satisfying (1). Phillips and Tai [31] managed to construct such coloring σ by Banaszczyk's Theorem and proved their result. Namely, a coloring satisfying (1) is construct-able.

Attempts to improve the result. We have seen how to show $|\mathcal{D}_{P,\sigma}(x)| < O(\sqrt{\log n})$. There is still a gap from showing $|\mathcal{D}_{P,\sigma}(x)| < O(1)$. We observe that the above argument aims at minimizing α such that the total failure probability $Ne^{-\Omega(\alpha^2)}$ is bounded by a constant. If we manage to make the factor N smaller, it helps setting α smaller and hence we can improve the result.

Recall that $N = O(n^2\sqrt{\log n}) = O(n \cdot n\sqrt{\log n})$ and the first factor n comes from the fact that P has n points and these n points could be widely spread out. Namely, we need at most n neighborhoods to cover all relevant grid points. What if all points in P lie inside a bounded region say $[-1, 1]$? In this case, we just need to consider *one* neighborhood to cover all relevant grid points. Nonetheless, we do not assume that they are in a bounded region and we take care of it in the following way. We partition \mathbb{R} into infinitely many bounded regions (say $\dots, [-3, -1], [-1, 1], [1, 3], \dots$) and assign each point in P to its corresponding region. Then, we construct a coloring on the points in each bounded region and each coloring is constructed independently. By triangle inequality, we have

$$|\mathcal{D}_{P,\sigma}(x)| \leq \sum |\mathcal{D}_{P_i,\sigma_i}(x)| \quad (2)$$

where each $P_i \subset P$ is the set of points in the same bounded region and σ_i is the coloring σ restricted on P_i .

If we manage to construct the colorings σ_i satisfying (1) then we will end up getting $|\mathcal{D}_{P,\sigma}(x)| < n_0 \cdot O(\alpha)$ where n_0 is the number of bounded regions that contain at least one point in P . However, n_0 can be as large as $O(n)$. To address this issue, we take the advantage of the assumption that all points in P_i are in a bounded region (say $[-1, 1]$). Since all points in P_i are in $[-1, 1]$ now and Gaussians decay exponentially, intuitively we should be able to construct a coloring σ_i that

$$|\mathcal{D}_{P_i,\sigma_i}(x)| < \alpha e^{-\frac{2}{3}x^2} \text{ for any } x \in \mathbb{R} \text{ with probability at least } 1 - O(e^{-\Omega(\alpha^2)}) \text{ for any } \alpha$$

if a coloring satisfying (1) is construct-able. It is because we can rewrite $|\mathcal{D}_{P_i, \sigma_i}(x)|$ as

$$|\mathcal{D}_{P_i, \sigma_i}(x)| = \left| \sum_{p \in P_i} \sigma_i(p) e^{-(x-p)^2} \right| = e^{-\frac{2}{3}x^2} \cdot \left| \sum_{p \in P_i} \sigma_i(p) e^{2p^2} e^{-\left(\frac{1}{\sqrt{3}}x - \sqrt{3}p\right)^2} \right| \tag{3}$$

and the expression in the RHS has a form similar to $\mathcal{D}_{P_i, \sigma_i}(x)$. The constant $\frac{2}{3}$ in the factor $e^{-\frac{2}{3}x^2}$ can be any constant between 0 and 1. The extra factor $e^{-\frac{2}{3}x^2}$ is crucial: when we plug the bound $\alpha e^{-\frac{2}{3}x^2}$ into (2), $|\mathcal{D}_{P, \sigma}(x)|$ is bounded by $O(1) \cdot O(\alpha)$ instead of $n_0 \cdot O(\alpha)$.

One minor issue here is that the failure probability is accumulated when we ensure all σ_i have the desired discrepancy. We fix this issue by turning the construction of each σ_i into a Las Vegas Algorithm. Namely, we check if each σ_i satisfies the desired discrepancy and repeat the construction if not.

Now, *if* we manage to construct a coloring σ such that: given $P \subset [-1, 1]$,

$$|\mathcal{D}_{P, \sigma}(x)| < \alpha e^{-\frac{2}{3}x^2} \text{ for any } x \in \mathbb{R} \text{ with probability at least } 1 - O(e^{-\Omega(\alpha^2)}) \text{ for any } \alpha \tag{4}$$

then we only need to consider *one* neighborhood to cover all relevant grid points when applying the union bound. We also preview here that (4) is the only property a coloring needs to show our result. From now on, we assume $P \subset [-1, 1]$. Even though (4) (the properties of the coloring σ we are looking for) is slightly different than (1) (what we stated in the beginning) because of the extra factor $e^{-\frac{2}{3}x^2}$, we can still perform a similar argument to prove that

$$|\mathcal{D}_{P, \sigma}(x)| < O(\sqrt{\log n}) e^{-\frac{2}{3}x^2} \text{ for all } x \in \mathbb{R} \text{ with probability at least } 1 - 1/10 \tag{5}$$

by arguing the slope of $\mathcal{D}_{P, \sigma}(x)$ is bounded by $O(n) e^{-\frac{2}{3}x^2}$ for any $x \in \mathbb{R}$.

Reusing the guarantees for $\mathcal{D}_{P, \sigma}$. Now, we look at the second factor $n\sqrt{\log n}$ in N . It turns out that we are not going to make this factor smaller. Instead, we will look at what guarantees this factor can give us and reuse these guarantees.

We further split $n\sqrt{\log n}$ into two parts: n and $\sqrt{\log n}$. Recall that the first part n comes from the configuration that the cell width of the grid is $\Omega(1/n)$ and the second part $\sqrt{\log n}$ comes from the configuration that we need to consider the neighborhood of radius $O(\sqrt{\log n})$ to cover all relevant grid points. However, we set up these two configurations *without taking σ into consideration*. As we mentioned before, if we have a coloring σ satisfying (4) then we have (5). Can we reuse this guarantee and exploit the coloring σ ? To answer this question, we first investigate the term $|\mathcal{D}_{P, \sigma}(x) - \mathcal{D}_{P, \sigma}(y)|$ for any $x, y \in \mathbb{R}$ and, by exploiting the structure of the Gaussians, we can prove

$$\left| \frac{\mathcal{D}_{P, \sigma}(x) - \mathcal{D}_{P, \sigma}(y)}{x - y} \right| < O(|\xi|) \cdot |\mathcal{D}_{P, \sigma}(\xi)| \tag{6}$$

for any $x \neq y$ where ξ is in between x and y . *The takeaway from this inequality is the slope of $\mathcal{D}_{P, \sigma}$ is bounded by $\mathcal{D}_{P, \sigma}$ itself.* It is how we can reuse our guarantees.

If we plug our guarantee (5) into (6), we can show that the slope of $\mathcal{D}_{P, \sigma}(x)$ for this σ is bounded by $O(\sqrt{\log n \log \log n}) e^{-\frac{2}{3}x^2}$ for any x within a radius of $O(\sqrt{\log \log n})$. For x that lies beyond a radius of $\Omega(\sqrt{\log \log n})$, we have

$$|\mathcal{D}_{P, \sigma}(x)| < O(\sqrt{\log n}) e^{-\frac{2}{3}x^2} < \frac{O(\sqrt{\log n})}{\Omega(\sqrt{\log n})} e^{-\frac{1}{3}x^2} < O(1) e^{-\frac{1}{3}x^2} < O(\sqrt{\log \log n}) e^{-\frac{1}{3}x^2} \tag{7}$$

63:8 Optimal Coreset for Gaussian Kernel Density Estimation

Note that the constant in the exponent becomes $\frac{1}{3}$ and it can be any constant smaller than $\frac{2}{3}$. If we have a coloring σ satisfying *additionally* that $|\mathcal{D}_{P,\sigma}(x)| < O(\sqrt{\log \log n})e^{-\frac{2}{3}x^2}$ for all x in the set of grid points that are in the grid of cell width $\Omega(1/\sqrt{\log n})$ (instead of $\Omega(1/n)$) and bounded within a radius of $O(\sqrt{\log \log n})$ (instead of $O(\sqrt{\log n})$), then we have

$$|\mathcal{D}_{P,\sigma}(x)| < O(\sqrt{\log \log n})e^{-\frac{1}{3}x^2} \text{ for all } x \in \mathbb{R}.$$

There is a caveat: to ensure the coloring σ satisfies the additional properties, we have to include more events in the union bound when invoking (4). In other words, the failure probability is now larger than $1/10$. Nonetheless, we improved the previous result to $|\mathcal{D}_{P,\sigma}(x)| < O(\sqrt{\log \log n})e^{-\frac{1}{3}x^2}$.

Hints of using induction. From the improvement we just made, it gives us a hint to refine the quality of our result by induction. One may notice the following pattern. Suppose we have a coloring σ satisfying

$$|\mathcal{D}_{P,\sigma}(x)| < \beta e^{-\kappa x^2} \text{ for all } x \in \mathbb{R} \quad (8)$$

for some β where κ is any constant between 0 and 1 (like $2/3$ before). Let S be the set of grid points that are in the grid of cell width $\Omega(1/\beta)$ and lie within a radius of $O(\sqrt{\log \beta})$. Note that $|S| = O(\beta\sqrt{\log \beta})$. If this coloring σ also satisfies that

$$|\mathcal{D}_{P,\sigma}(x)| < O(\sqrt{\log \beta})e^{-\kappa x^2} \text{ for all } x \in S \quad (9)$$

then we can modify the previous argument in the following way. From (8) and (6), we have the absolute value of the slope of $\mathcal{D}_{P,\sigma}$ at any point within a radius of $O(\sqrt{\log \beta})$ is bounded by $O(\beta\sqrt{\log \beta})e^{-\kappa x^2}$. From an argument similar to (7), we also have $|\mathcal{D}_{P,\sigma}(x)| < O(\sqrt{\log \beta})e^{-\kappa' x^2}$ for all x that lies beyond a radius of $\Omega(\sqrt{\log \beta})$ where κ' is any constant between 0 and κ (like $1/3$ before). We combine them with (9) and it implies

$$|\mathcal{D}_{P,\sigma}(x)| < O(\sqrt{\log \beta})e^{-\kappa' x^2} \text{ for all } x \in \mathbb{R}. \quad (10)$$

If we take (5) as the base step and the implication from (8) to (10) as the inductive step, we should expect

$$|\mathcal{D}_{P,\sigma}(x)| < O(1)e^{-\frac{1}{3}x^2} \text{ for all } x \in \mathbb{R}.$$

after $O(\log^* n)$ inductive steps.

As we mentioned before, we also need to keep track of the failure probability and the exponent κ in the factor $e^{-\kappa x^2}$. We first deal with the failure probability. In each inductive step, we need extra guarantees on the set of grid points of a smaller size (i.e. (9) when invoking (4)). Hence, the total failure probability is a sum of $O(\log^* n)$ failure probabilities in each inductive step. We can set these $O(\log^* n)$ failure probabilities to be a geometric sequence such that the total failure probability is a constant. The other issue is the exponent. We can again make this exponent decrease from $2/3$ to $1/3$ geometrically as it proceeds in the inductive steps. In each inductive step, we need to set α in (4) larger than what we stated earlier accordingly when invoking (4) in the union bound. Nonetheless, we eventually prove that $|\mathcal{D}_{P,\sigma}(x)| < O(1)e^{-\frac{1}{3}x^2}$ for all $x \in \mathbb{R}$ with probability $1/2$.

Construction of the coloring. It all boils down to the problem of how to construct a coloring σ satisfying (4). Namely, given a point set $P \subset [-1, 1]$,

$$|\mathcal{D}_{P,\sigma}(x)| < \alpha e^{-\frac{2}{3}x^2} \text{ for any } x \in \mathbb{R} \text{ with probability at least } 1 - O(e^{-\Omega(\alpha^2)}) \text{ for any } \alpha.$$

We introduced Banaszczyk’s Theorem (Theorem 3) before and if we can rewrite (4) as the inner product form shown in Theorem 3 then we can apply the algorithm in Theorem 3. As we mentioned in (3), we first rewrite

$$|\mathcal{D}_{P,\sigma}(x)| = \left| \sum_{p \in P} \sigma(p) e^{-(x-p)^2} \right| = e^{-\frac{2}{3}x^2} \cdot \left| \sum_{p \in P} \sigma(p) e^{2p^2} e^{-\left(\frac{1}{\sqrt{3}}x - \sqrt{3}p\right)^2} \right|.$$

and hence we can ease the notation by dropping the factor $e^{-\frac{2}{3}x^2}$. Namely, we need a coloring σ such that, given a point set $P \subset [-1, 1]$,

$$\left| \sum_{p \in P} \sigma(p) e^{2p^2} e^{-\left(\frac{1}{\sqrt{3}}x - \sqrt{3}p\right)^2} \right| < \alpha \text{ for any } x \in \mathbb{R}$$

with probability at least $1 - O(e^{-\Omega(\alpha^2)})$ for any α . Since the Gaussian kernel is a positive-definite kernel, it implies that the term $e^{-\left(\frac{1}{\sqrt{3}}x - \sqrt{3}p\right)^2}$ can be rewritten as $\langle u^{(\frac{1}{\sqrt{3}}x)}, u^{(\sqrt{3}p)} \rangle$ where $u^{(\cdot)}$ is a vector such that $\langle u^{(s)}, u^{(t)} \rangle = e^{-(s-t)^2}$ for any $s, t \in \mathbb{R}$. It is worth noting that $\|u^{(s)}\|^2 = \langle u^{(s)}, u^{(s)} \rangle = e^{-(s-s)^2} = 1$ for any $s \in \mathbb{R}$. Hence, we further rewrite (4) as: given a point set $P \subset [-1, 1]$,

$$|\langle u^{(\frac{1}{\sqrt{3}}x)}, \Sigma \rangle| < \alpha \text{ for any } x \in \mathbb{R} \text{ with probability at least } 1 - O(e^{-\Omega(\alpha^2)}) \text{ for any } \alpha$$

where $\Sigma = \sum_{p \in P} \sigma(p) e^{2p^2} u^{(\sqrt{3}p)}$. It is the inner product form we are looking for in order to apply the algorithm in Theorem 3. Recall that the norms of the input vectors and the query vectors in Banaszczyk’s Theorem are required to be not larger than 1. We check that the norm of the query vector $\|u^{(\frac{1}{\sqrt{3}}x)}\| = 1$ and the norm of the input vector $\|e^{2p^2} u^{(\sqrt{3}p)}\| = O(1)$ since we assume that $P \subset [-1, 1]$. Karnin and Liberty [21] assumed *both* the point set P and the query x lie within a constant radius because their result stops short of handling the norms of these vectors when using Banaszczyk’s Theorem. If we take $\frac{e^{2p^2} u^{(\sqrt{3}p)}}{\|e^{2p^2} u^{(\sqrt{3}p)}\|}$ as the input vectors, we can apply the algorithm in Theorem 3 to construct the desired coloring.

4 Proofs

In this section, we will show how to construct an ε -coreset via discrepancy theory. From now on, we assume that d is a constant. The log function in this paper is base e . Also, we define the following notations. Let $\text{Grid}_d(\gamma) \subset \mathbb{R}^d$ be an infinite lattice grid of cell width γ , i.e. $\{(\gamma i_1, \dots, \gamma i_d) \mid i_1, \dots, i_d \text{ are integers}\}$. Denote $B_\infty^d(r) = \{x \mid |x_j| < r \text{ for } j = 1, \dots, d\}$ to be a ℓ_∞ -ball of radius r . We define a decreasing sequence n_i in the following way: $n_0 = \log^2 n$, $n_1 = \sqrt{3} \log n + 3$ and $n_{i+1} = \sqrt{3} \cdot 2^{\ell(n)-i} \log n_i$ for $i = 1, \dots, \ell(n) - 1$. Here, $\ell(n) + 3$ is the smallest integer k that $\text{ilog}(k, n) < 0$ where $\text{ilog}(k, n) = \log \dots \log n$ (there are k log functions) and it is easy to see that $\ell(n) = O(\log^* n)$. For $i = 0, \dots, \ell(n) - 1$, denote $S_i = \text{Grid}_d(\frac{1}{C_0 n_i}) \cap [-n_{i+1}, n_{i+1}]^d = \text{Grid}_d(\frac{1}{C_0 n_i}) \cap B_\infty^d(n_{i+1})$ where C_0 is a sufficiently large constant. Namely, S_i is a bounded lattice grid and its size is at most $(2C_0 n_i n_{i+1})^d$. Note that S_i may be interpreted as a subset of S_0 but, for clarity, we still view them as different sets. Throughout this section, the absolute constants C_0, C_1, C are unchanged. Also, C is larger than C_1 and C_1 is larger than C_0 .

4.1 Useful lemmas

Before we go into the main proof, we first present some important observations.

► **Lemma 5.** *Suppose $P \subset B_\infty^d(1)$ be a point set of size n and σ is a coloring on P . Then, we have*

$$\sup_{x \in B_\infty^d(\sqrt{3 \log n} + 3)} \left| \sum_{p \in P} \sigma(p) e^{2\|p\|^2} e^{-\frac{1}{3}\|x-3p\|^2} \right| \leq 4 \cdot \sup_{s \in S_0} \left| \sum_{p \in P} \sigma(p) e^{2\|p\|^2} e^{-\frac{1}{3}\|s-3p\|^2} \right| + 7$$

where $S_0 = \text{Grid}_d(w) \cap [-\sqrt{3 \log n} - 3, \sqrt{3 \log n} + 3]^d = \text{Grid}_d(w) \cap B_\infty^d(\sqrt{3 \log n} + 3)$ with $w = \frac{1}{C_0 \log^2 n}$. Here, $\text{Grid}_d(\gamma) = \{(\gamma i_1, \dots, \gamma i_d) \mid i_1, \dots, i_d \text{ are integers}\} \subset \mathbb{R}^d$ is an infinite lattice grid.

The main technique in Lemma 5 is expanding the expression by Taylor expansion. Then, by truncating the Taylor expansion with a finite number of terms, one can bound the derivatives of the expression by using Markov Brother's inequality (Theorem 4). Since the width of the grid cell in S_0 depends on the number of terms in the Taylor expansion, we need to argue that a small number of terms suffices to bound the error.

► **Lemma 6.** *Given a coloring σ . For any $x, s \in \mathbb{R}^d$ such that $|x_j| \leq |s_j|$ for all $j = 1, 2, \dots, d$, we have*

$$\begin{aligned} & \left| \sum_{p \in P} \sigma(p) e^{-\|x-p\|^2} - \sum_{p \in P} \sigma(p) e^{-\|s-p\|^2} \right| \\ & \leq (\|s\|^2 - \|x\|^2) \left| \sum_{p \in P} \sigma(p) e^{-\|x-p\|^2} \right| + 2 \sum_{j=1}^d |s_j - x_j| \cdot \left| \sum_{p \in P} \sigma(p) e^{-\|\xi^{(j)}-p\|^2} \right| \end{aligned}$$

where $\xi^{(j)} = (x_1, \dots, x_{j-1}, \xi_j, s_{j+1}, \dots, s_d)$ for some ξ_j between $|x_j|$ and $|s_j|$.

In the inductive step, the main observation is the absolute difference of the discrepancy objective at two different points, $|\mathcal{D}_{P,\sigma}(x) - \mathcal{D}_{P,\sigma}(y)|$, can be bounded by the discrepancy objective itself. Lemma 6 is the lemma providing the key inequality to perform the inductive steps.

Finally, we also show the asymptotic bound of the recurrence equation n_i in Lemma 7.

► **Lemma 7.** *Let $n_0 = \log^2 n$, $n_1 = \sqrt{3 \log n} + 3$ and $n_{i+1} = \sqrt{3 \cdot 2^{\ell(n)-i} \log n_i}$ for $i = 1, \dots, \ell(n) - 1$. Then, $n_{\ell(n)} = O(1)$. Recall that $\ell(n) + 3$ is the smallest integer k that $\text{ilog}(k, n) < 0$.*

4.2 Base step

Recall that the definition of $\mathcal{D}_{P,\sigma}(x)$ is $\sum_{p \in P} \sigma(p) e^{-\|x-p\|^2}$. Lemma 8 shows that if a coloring σ satisfies that $|\mathcal{D}_{P,\sigma}(x)|$ is small for all x in a finite subset (which is a grid) of \mathbb{R}^d , then the coloring σ also satisfies that $|\mathcal{D}_{P,\sigma}(x)|$ is small for all $x \in \mathbb{R}^d$. Note that we still haven't provided the detail on how to find such coloring and we will do it in the full algorithm.

► **Lemma 8.** *Suppose $P \subset B_\infty^d(1)$. Given a coloring σ such that, for all $s' \in S_0 = \text{Grid}_d(w) \cup B_\infty^d(n_1)$ where $w = \frac{1}{C_0 \log^2 n} = \frac{1}{C_0 n_0}$ is the same w shown in Lemma 5 and $n_1 = \sqrt{3 \log n} + 3$,*

$$|\mathcal{D}_{P,\sigma}(s')| = \left| \sum_{p \in P} \sigma(p) e^{-\|s'-p\|^2} \right| < C_1 n_1 e^{-\frac{2}{3}\|s'\|^2}$$

Then, we have, for all $x \in \mathbb{R}^d$,

$$|\mathcal{D}_{P,\sigma}(x)| = \left| \sum_{p \in P} \sigma(p)e^{-\|x-p\|^2} \right| < Cn_1e^{-\frac{2}{3}\|x\|^2}.$$

Here, C, C_1, C_0 are sufficiently large constant depending on d only.

We make a short remark here. One might notice that Lemma 8 states $w = \frac{1}{C_0 \log^2 n}$ while it is sufficient to set $w = \Omega(\frac{1}{n})$ as suggested in Section 3. As we mentioned before, our final algorithm is a Las Vegas algorithm and hence we need to check if the output coloring has the desired discrepancy. We check it by enumerating the relevant grid points and computing the discrepancy at them. Making w larger reduces the size of the grid and hence improves the running time. Nonetheless, $w = \frac{1}{C_0 \log^2 n} = \Omega(\frac{1}{n})$ and hence it doesn't change the logic.

4.3 Inductive step

Lemma 9 suggests that if a coloring σ satisfies that $|\mathcal{D}_{P,\sigma}(x)|$ is small for all $x \in \mathbb{R}^d$, then the coloring σ satisfies that the absolute difference $|\mathcal{D}_{P,\sigma}(x) - \mathcal{D}_{P,\sigma}(s)|$ is also small for any two close points x, s within a certain region. It is achieved by the observation that the slope of $\mathcal{D}_{P,\sigma}$ can be bounded by $\mathcal{D}_{P,\sigma}$ itself in magnitude.

► **Lemma 9.** Suppose $P \subset B_\infty^d(1)$. Let $D_i = C \cdot \frac{5}{4}(1 - \frac{1}{5^i})$ and $I_i = \frac{1}{3} + \frac{1}{3}(1 - \frac{1}{2^{\ell(n)-i}})$. Given a coloring σ such that, for all $x \in \mathbb{R}^d$,

$$|\mathcal{D}_{P,\sigma}(x)| = \left| \sum_{p \in P} \sigma(p)e^{-\|x-p\|^2} \right| < D_i \cdot n_i e^{-I_i \|x\|^2}.$$

If $x \in B_\infty^d(n_{i+1})$, then

$$|\mathcal{D}_{P,\sigma}(x) - \mathcal{D}_{P,\sigma}(s)| = \left| \sum_{p \in P} \sigma(p)e^{-\|x-p\|^2} - \sum_{p \in P} \sigma(p)e^{-\|s-p\|^2} \right| \leq \frac{1}{5} D_i \cdot n_{i+1} e^{-I_i \|x\|^2}.$$

where $s \in S_i$ is the closest point to x that $|s_j| > |x_j|$ for all $j = 1, 2, \dots, d$.

Similar to Lemma 8, Lemma 10 shows that if a coloring σ satisfies that $|\mathcal{D}_{P,\sigma}(x)|$ is small for all x in a finite subset (which is a grid) of \mathbb{R}^d , then the coloring σ also satisfies that $|\mathcal{D}_{P,\sigma}(x)|$ is small for all $x \in \mathbb{R}^d$. The only difference is that we can take the advantage of the discrepancy guarantee from the previous iterations.

► **Lemma 10.** Suppose $P \subset B_\infty^d(1)$. Recall that $D_i = C \cdot \frac{5}{4}(1 - \frac{1}{5^i})$ and $I_i = \frac{1}{3} + \frac{1}{3}(1 - \frac{1}{2^{\ell(n)-i})}$ which is the same definition as in Lemma 9. Given a coloring σ such that, for all $s' \in S_i$,

$$\left| \sum_{p \in P} \sigma(p)e^{-\|s'-p\|^2} \right| < C_1 n_{i+1} e^{-\frac{2}{3}\|s'\|^2}$$

and, for all $x \in \mathbb{R}^d$,

$$\left| \sum_{p \in P} \sigma(p)e^{-\|x-p\|^2} \right| \leq D_i \cdot n_i e^{-I_i \|x\|^2}.$$

Then, we have, for all $x \in \mathbb{R}^d$,

$$\left| \sum_{p \in P} \sigma(p)e^{-\|x-p\|^2} \right| < D_{i+1} \cdot n_{i+1} e^{-I_{i+1} \|x\|^2}.$$

Here, C, C_1 are sufficiently large constants.

4.4 Full algorithm

For now, we still assume that $P \subset B_\infty^d(1)$. Now, we can apply the algorithm in Theorem 3 to construct our coloring σ that produces a low discrepancy, $|\mathcal{D}_{P,\sigma}(x)|$, for all $x \in \mathbb{R}^d$. Recall that $\ell(n) + 3$ is the smallest integer k that $\text{ilog}(k, n) < 0$. Also, we defined n_i before such that $n_0 = \log^2 n$, $n_1 = \sqrt{3 \log n} + 3$ and $n_{i+1} = \sqrt{3 \cdot 2^{\ell(n)-i} \log n_i}$.

► **Lemma 11.** *Assuming $P \subset B_\infty^d(1)$. Given a set of vectors V_P defined as follows.*

$$V_P = \left\{ \frac{1}{\sqrt{1+e^{4d}}} \left(\frac{1}{v^{(p)} e^{2\|p\|^2}} \mid p \in P \right) \right\}$$

such that $\langle v^{(p)}, v^{(q)} \rangle = e^{-3\|p-q\|^2}$ for any $p, q \in P$. Then, by taking V_P as the input, the algorithm in Theorem 3 constructs a coloring σ on P such that

$$\left| \sum_{p \in P} \sigma(p) e^{-\|x-p\|^2} \right| < C \cdot \frac{5}{4} \cdot n_{\ell(n)} e^{-\frac{1}{3}\|x\|^2}$$

for all $x \in \mathbb{R}^d$ and $|\sum_{p \in P} \sigma(p)| \leq C$ with probability at least $\frac{1}{2}$.

Recall that we eventually would like to use the halving technique to construct our ε -coreset. To use the halving technique, we need to ensure that half of the points in P are $+1$ and the other half are -1 . In Lemma 11, the 1s concatenated on top of the vectors $v^{(p)} e^{2\|p\|^2}$ in V_P ensure the coloring has the above property.

► **Lemma 12.** *Assuming $P \subset B_\infty^d(1)$. There is an efficient algorithm that constructs a coloring σ such that $|\sum_{p \in P} \sigma(p) e^{-\|x-p\|^2}| = O(n_{\ell(n)} e^{-\frac{1}{3}\|x\|^2})$ for all $x \in \mathbb{R}^d$ and half of points are assigned $+1$ with probability at least $\frac{1}{2}$.*

■ **Algorithm 1** Construction of the coloring.

input: a point set $P \subset \mathbb{R}^d$

- 1: initialize $Q_g = \emptyset$ for all $g \in \text{Grid}_d(2)$
 - 2: **for** each $p \in P$ **do**
 - 3: insert p into Q_g where $g \in \text{Grid}_d(2)$ is the closest point to p .
 - 4: **for** each non-empty Q_g **do**
 - 5: construct a collection V_g of vector $\left\{ \frac{1}{\sqrt{1+e^{4d}}} \left(\frac{1}{v^{(p)} e^{2\|p\|^2}} \mid p \in Q_g \right) \right\}$ such that $\langle v^{(p)}, v^{(q)} \rangle = e^{-3\|p-q\|^2}$ for any $p, q \in Q_g$
 - 6: use V_g as the input and run the algorithm in Theorem 3 to obtain a coloring σ_g on Q_g
 - 7: check if σ_g satisfies the conditions in Lemma 8 and Lemma 10 and repeat line 6 if not
 - 8: flip the color of any points such that half of points in Q_g are colored $+1$.
 - 9: **return** a coloring $\sigma : P \rightarrow \{-1, +1\}$ such that $\sigma(p) = \sigma_g(p)$ when $p \in Q_g$
-

We can now remove the assumption of $P \subset B_\infty^d(1)$. Algorithm 1 is a Las Vegas algorithm that constructs a coloring on the input point set P . We can now show how to construct a coloring such that the discrepancy is small. Recall that we defined $\text{Grid}_d(\gamma) = \{(\gamma i_1, \dots, \gamma i_d) \mid i_1, \dots, i_d \text{ are integers}\} \subset \mathbb{R}^d$ to be an infinite lattice grid. The idea of Algorithm 1 is that we first decompose the entire \mathbb{R}^d into infinitely many ℓ_∞ -balls of radius 1. Then, we partition our input P such that each point $p \in P$ lies in some ℓ_∞ -ball. For each non-empty ℓ_∞ -ball, run the algorithm in Theorem 3 to construct a coloring with the desired discrepancy by Lemma 12. Finally, we argue that there is an extra constant factor in the final discrepancy.

► **Lemma 13.** *Suppose $P \subset \mathbb{R}^d$ be a point set of size n . Then, Algorithm 1 constructs a coloring σ on P efficiently such that $\sup_{x \in \mathbb{R}^d} |\sum_{p \in P} \sigma(p) e^{-\|x-p\|^2}| = O(1)$ and half of the points in P are colored $+1$.*

One can first perform random sampling [24] before running Algorithm 1 such that the input size $n = O(\frac{1}{\varepsilon^2})$. Finally, by the standard halving technique, we have the following theorem.

► **Theorem 14 (Restated Theorem 1).** *Suppose $P \subset \mathbb{R}^d$ be a point set of size n . Let $\bar{\mathcal{G}}_P$ be the Gaussian kernel density estimate of P , i.e. $\bar{\mathcal{G}}_P(x) = \frac{1}{|P|} \sum_{p \in P} e^{-\|x-p\|^2}$ for any $x \in \mathbb{R}^d$. For a fixed constant d , there is an algorithm that constructs a subset $Q \subset P$ of size $O(\frac{1}{\varepsilon})$ such that $\sup_{x \in \mathbb{R}^d} |\bar{\mathcal{G}}_P(x) - \bar{\mathcal{G}}_Q(x)| < \varepsilon$ and has a polynomial running time in n .*

5 Conclusion and discussion

In this paper, we studied the question of constructing coresets for kernel density estimates. We proved that the Gaussian kernel has an ε -coreset of the optimal size $O(1/\varepsilon)$ when d is a constant. This coreset can be constructed efficiently. We leveraged Banaszczyk's Theorem to construct a coloring such that the kernel discrepancy is small. Then, we constructed an ε -coreset of the desired size via the halving technique.

Some open problems in discrepancy theory, such as Tusnády's Problem, have an issue that an extra factor shows up when we generalize the result from the case of $d = 1$ to the case of larger d . A previous result of our problem is reducing our problem to Tusnády's Problem. It turns out that, if $d = 1$, the trivial solution gives the optimal result. Unfortunately, it cannot be generalized to the higher dimensional case. Our new induction analysis combining with Banaszczyk's Theorem provides a non-trivial perspective even when $d = 1$. Hence, it might open up a possibility of improving the results on these open problems.

Even though the Gaussian kernel is a major class of kernels in most of applications, it would be interesting to investigate similar results on other kernel settings such as the Laplace kernel. Our approach exploits the properties of the Gaussian kernel such as factoring out the $e^{-\Omega(\|x\|^2)}$ factor while maintaining the positive-definiteness. Generalizing the result to a broader class of kernels might require deeper understandings of the properties that other kernels share with the Gaussian kernel.

In some applications, the input data might be in the high dimensional space. Our result assumes that d is a constant. Note that one of the previous results is sub-optimal in terms of ε but is optimal in terms of d . The dependence on d in our result is exponential which we might want to avoid in the high dimensional case. Hence, improving the dependence on d to polynomial is also interesting because it would be more practical in some applications.

References

- 1 Pankaj K Agarwal, Sariel Har-Peled, Haim Kaplan, and Micha Sharir. Union of random minkowski sums and network vulnerability analysis. *Discrete & Computational Geometry*, 52(3):551–582, 2014.
- 2 Christoph Aistleitner, Dmitriy Bilyk, and Aleksandar Nikolov. Tusnády's problem, the transference principle, and non-uniform qmc sampling. In *International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 169–180. Springer, 2016.
- 3 Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.
- 4 Wojciech Banaszczyk. Balancing vectors and gaussian measures of n-dimensional convex bodies. *Random Structures & Algorithms*, 12(4):351–360, 1998.

- 5 Nikhil Bansal, Daniel Dadush, Shashwat Garg, and Shachar Lovett. The gram-schmidt walk: a cure for the banaszczyk blues. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 587–597, 2018.
- 6 Nikhil Bansal and Shashwat Garg. Algorithmic discrepancy beyond partial coloring. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 914–926, 2017.
- 7 József Beck. Roth’s estimate of the discrepancy of integer sequences is nearly sharp. *Combinatorica*, 1(4):319–325, 1981.
- 8 Jon Louis Bentley and James B Saxe. Decomposable searching problems i: Static-to-dynamic transformation. *J. algorithms*, 1(4):301–358, 1980.
- 9 Moses Charikar, Michael Kapralov, Navid Nouri, and Paris Siminelakis. Kernel density estimation through density constrained near neighbor search. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 172–183. IEEE, 2020.
- 10 Frédéric Chazal, Brittany Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, Alessandro Rinaldo, and Larry Wasserman. Robust topological inference: Distance to a measure and kernel distance. *The Journal of Machine Learning Research*, 18(1):5845–5884, 2017.
- 11 Bernard Chazelle. *The discrepancy method: randomness and complexity*. Cambridge University Press, 2001.
- 12 Bernard Chazelle and Jiří Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *Journal of Algorithms*, 21(3):579–597, 1996.
- 13 Kenneth L Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):1–30, 2010.
- 14 Luc Devroye and László Györfi. *Nonparametric Density Estimation: The L_1 View*. Wiley, 1984.
- 15 Bernd Gärtner and Martin Jaggi. Coresets for polytope distance. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 33–42, 2009.
- 16 Leslie Greengard and John Strain. The fast gauss transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991.
- 17 Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.
- 18 Mingxuan Han, Michael Matheny, and Jeff M Phillips. The kernel spatial scan statistic. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 349–358, 2019.
- 19 Phillips Jeff and Tai Wai Ming. The gaussiansketch for almost relative error kernel distance. In *International Conference on Randomization and Computation (RANDOM)*, 2020.
- 20 Sarang Joshi, Raj Varma Kommaraji, Jeff M Phillips, and Suresh Venkatasubramanian. Comparing distributions and shapes using the kernel distance. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 47–56, 2011.
- 21 Zohar Karnin and Edo Liberty. Discrepancy, coresets, and sketches in machine learning. In *Conference on Learning Theory*, pages 1975–1993, 2019.
- 22 Simon Lacoste-Julien, Fredrik Lindsten, and Francis Bach. Sequential kernel herding: Frank-wolfe optimization for particle filtering. In *Artificial Intelligence and Statistics*, pages 544–552, 2015.
- 23 Jasper CH Lee, Jerry Li, Christopher Musco, Jeff M Phillips, and Wai Ming Tai. Finding an approximate mode of a kernel density estimate. In *29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- 24 David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Iliya Tolstikhin. Towards a learning theory of cause-effect inference. In *International Conference on Machine Learning*, pages 1452–1461, 2015.
- 25 AA Markov. On a question of di mendelev, zap. *Petersburg Akad. Nauk*, 62:1–24, 1889.

- 26 Jiri Matousek. *Geometric discrepancy: An illustrated guide*, volume 18. Springer Science & Business Media, 2009.
- 27 Jiří Matoušek, Aleksandar Nikolov, and Kunal Talwar. Factorization norms and hereditary discrepancy. *International Mathematics Research Notices*, 2020(3):751–780, 2020.
- 28 Jeff M Phillips. Algorithms for ε -approximations of terrains. In *International Colloquium on Automata, Languages, and Programming*, pages 447–458. Springer, 2008.
- 29 Jeff M Phillips. ε -samples for kernels. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1622–1632. SIAM, 2013.
- 30 Jeff M Phillips and Wai Ming Tai. Improved coresets for kernel density estimates. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2718–2727. SIAM, 2018.
- 31 Jeff M Phillips and Wai Ming Tai. Near-optimal coresets of kernel density estimates. *Discrete & Computational Geometry*, pages 1–21, 2019.
- 32 Jeff M Phillips, Bei Wang, and Yan Zheng. Geometric inference on kernel density estimates. In *31st International Symposium on Computational Geometry (SoCG 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.
- 33 Alessandro Rinaldo, Larry Wasserman, et al. Generalized density clustering. *The Annals of Statistics*, 38(5):2678–2722, 2010.
- 34 Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- 35 David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.
- 36 Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.
- 37 Joel Spencer. Six standard deviations suffice. *Transactions of the American mathematical society*, 289(2):679–706, 1985.
- 38 Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010.
- 39 Grace Wahba et al. Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods-Support Vector Learning*, 6:69–87, 1999.
- 40 Yan Zheng and Jeff M Phillips. L_∞ error and bandwidth selection for kernel density estimates of large data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1533–1542, 2015.
- 41 Shaofeng Zou, Yingbin Liang, H Vincent Poor, and Xinghua Shi. Unsupervised nonparametric anomaly detection: A kernel method. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 836–841. IEEE, 2014.

GPU Computation of the Euler Characteristic Curve for Imaging Data

Fan Wang ✉

Stony Brook University, NY, US

Hubert Wagner ✉

University of Florida, Gainesville, FL, US

Chao Chen ✉

Stony Brook University, NY, US

Abstract

Persistent homology is perhaps the most popular and useful tool offered by topological data analysis – with point-cloud data being the most common setup. Its older cousin, the Euler characteristic curve (ECC) is less expressive – but far easier to compute. It is particularly suitable for analyzing imaging data, and is commonly used in fields ranging from astrophysics to biomedical image analysis. These fields are embracing GPU computations to handle increasingly large datasets.

We therefore propose an optimized GPU implementation of ECC computation for 2D and 3D grayscale images. The goal of this paper is twofold. First, we offer a practical tool, illustrating its performance with thorough experimentation – but also explain its inherent shortcomings. Second, this simple algorithm serves as a perfect backdrop for highlighting basic GPU programming techniques that make our implementation so efficient – and some common pitfalls we avoided. This is intended as a step towards a wider usage of GPU programming in computational geometry and topology software. We find this is particularly important as geometric and topological tools are used in conjunction with modern, GPU-accelerated machine learning frameworks.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Combinatorial algorithms

Keywords and phrases topological data analysis, Euler characteristic, Euler characteristic curve, Betti curve, persistent homology, algorithms, parallel programming, algorithm engineering, GPU programming, imaging data

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.64

Related Version *Full Version*: <https://arxiv.org/abs/2203.09087>

Supplementary Material *Software (Source Code)*: https://github.com/TopoXLab/GPU_ECC_SoCG2022; archived at [swh:1:dir:5f915660625457e3fbb99aeb77a7160385580560](https://swh.1.dir:5f915660625457e3fbb99aeb77a7160385580560)

Funding This work was partially supported by grants NSF IIS-1909038 and CCF-1855760.

1 Introduction

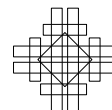
Describing the shape of data is the tenet of topological data analysis (TDA) – and at its heart lies the idea of studying data across scales. Instead of characterizing the shape at a fixed scale – we measure its evolution. A filtration encodes this evolution and thus becomes an object of primary interest. Depending on the type of data, an appropriate filtration is used: Alpha-shape filtration for point-cloud data embedded in three dimensional space; Vietoris–Rips filtration for high dimensional metric data expressed by pairwise distances; cubical filtration for two- or three-dimensional grayscale imaging data. This paper focuses on imaging data, in which TDA methods have shown promise in recent years [21, 22, 24, 10].



© Fan Wang, Hubert Wagner, and Chao Chen;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 64; pp. 64:1–64:16
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Persistent homology is perhaps the most powerful topological descriptor applied to such filtrations – and it proves especially useful in conjunction with modern deep learning (DL) methods. However, integration of persistent homology with DL methods remains far from seamless – despite significant progress, computing persistent homology takes significant amount of time and resources for practical datasets. This is in contrast with modern learning pipelines which often employ simple, highly optimized computations. In particular, many neural network architectures are realized fully on graphical processing units (GPUs) attaining massively parallel processing; the same applies to modern large-scale simulations. Existing software for persistent homology is not at this level of advancement, at least not for imaging data. We mention that the recent GPU implementation by Zhang et al. [25] is in the context of Vietoris–Rips filtrations coming from point-cloud data and cannot handle imaging data.

In view of the above, we turn our attention to a simpler – but still expressive – topological descriptor, namely the Euler characteristic curve (ECC). ECC has an excellent track record in providing relevant topological information in various imaging applications [4, 2, 5] – we elaborate on this in Section 3. More importantly, we demonstrate that we can compute ECC at extremely fast speed – for example we can process a 3D image of size 512^3 30 times per second. We also managed to implement a streaming strategy, which allows us to handle huge images of 4096^3 and beyond – despite the limited GPU memory. The above points imply that a truly seamless integration with modern image processing pipelines is achievable. Overall, we hope to impact the following field.

Machine learning. We are particularly interested in incorporating ECC computation into machine learning frameworks, e.g., convolutional neural networks (CNNs) for computer vision [12], biomedical image processing [19] or computational astrophysics [16]. In these contexts, ECC can be used as topological features for prediction models.

Contributions. The main technical contribution of this paper is a streaming GPU implementation of ECC computation for imaging data. While the underlying algorithm is very simple, our contribution lies in the implementation carefully tuned to modern GPUs. In particular, when adapting computation into massive parallelism, we need to carefully design the implementation so that the limited GPU memory resources can be exploited in the most efficient manner.

2 Background

2.1 Images as cubical filtrations

The input to our algorithm is a d -dimensional grayscale image, by which we simply mean a d -dimensional array of real values. Individual elements are called pixels (in 2D) and voxels (in 3D and above), and we collectively call them voxels. One common operation is *thresholding*, which selects the subset of voxels not exceeding a certain threshold t . To talk about the topology of a sequence of thresholdings, we impose more structures on the data.

To this end we follow [11]. First, we define an elementary interval as either $[k, k + 1]$, or a degenerate interval $[k, k]$, for an integer k . An elementary (cubical) cell is a product of d elementary intervals, and its dimension is the number of non-degenerate intervals entering its product. This way we can talk about vertices, edges, squares, cubes etc as cells of dimension 0, 1, 2, 3 etc. We say that cell a is a face of cell b iff $a \subset b$, or a coface if $b \subset a$. Now, we associate the input values with the top dimensional cells, which we call *voxels*. Finally we extend the values from voxels to all lower dimensional cells: each cell inherits the minimum value of its top-dimensional cofaces. The thresholding of the image at value t is now a *cubical complex*, $K_{\leq t}$; the nested sequence of these complexes form a *cubical filtration* indexed by t .

■ **Algorithm 1** Sequential computation of the VCEC.

Require: I : an input image

Ensure: $VCEC$: the vector of changes in the Euler characteristic.

- 1: initialize $VCEC$ as an empty array
 - 2: **for all** voxels v in I **do**
 - 3: **for all** faces c of v **do**
 - 4: **if** c was introduced by v **then**
 - 5: $VCEC[\text{value of } v \text{ in } I] \leftarrow VCEC[\text{value of } v \text{ in } I] + (-1)^{\text{dimension of face } c}$
-

2.2 The Euler characteristic curve

With the above setup, we can define the Euler characteristic curve of a cubical filtration as the sequence

$$ECC_i = \chi(K_{\leq t_i}) = \sum_j (-1)^j c_j(K_{\leq t_i}) = \sum_j (-1)^j \beta_j(K_{\leq t_i}) \quad (1)$$

where t_i is the i -th smallest grayscale value in the image, $c_j(\cdot)$ counts the j -dimensional cells, and $\beta_j(\cdot)$ is the j -dimensional Betti number. The last equality comes from the Euler–Poincaré formula and ties ECC with the topology of the space.

We only mention that the Betti numbers are the ranks of the *cubical homology groups* [11] of the cubical complex $K_{\leq t}$. For three dimensional complexes, the Betti numbers count the number of connected components, tunnels and voids in an object. Therefore, the ECC mixes up the numbers of topological features at each threshold. We can also define *persistent homology* [6] in this setup [20]. Fig. 1 illustrates the relationship between persistent homology, the Betti curves and the ECC.

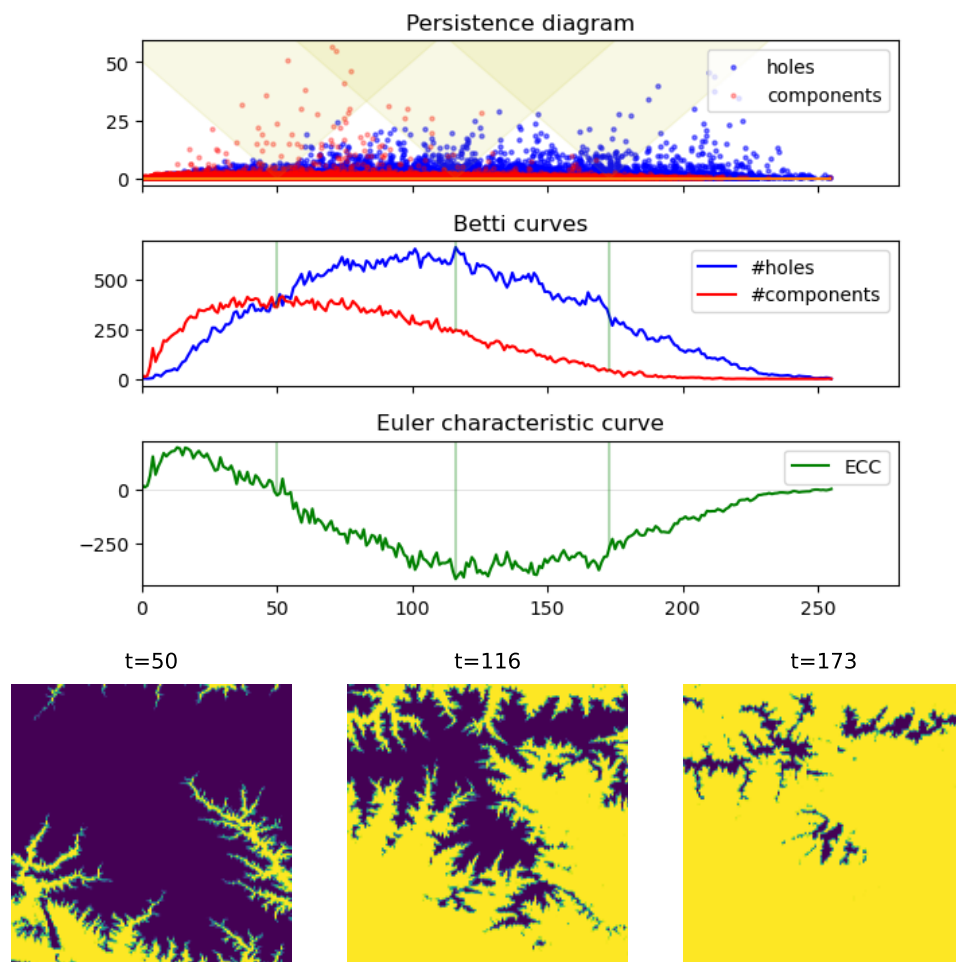
2.3 ECC computation

A naive algorithm for ECC explicitly computes the Euler characteristic (EC) at each threshold. This results in time complexity of $O(mn)$, where m is the number of unique values in the image, and n the number of voxels. We assume the dimension of the image is a constant.

An algorithm by Snidaro and Foresti [18] was the first offering $O(n)$ complexity, but is quite complicated and hard to generalize beyond 2D. Our approach is based on a much simpler algorithm [8], which interprets an image as a cubical filtration.

Tracking the VCEC. The main idea is to compute the “Vector of Changes in the Euler Characteristic” (VCEC), namely a sequence of length m such that $VCEC_0 = ECC_0$ and $VCEC_i = ECC_i - ECC_{i-1}$, for $0 < i < m$. Since $ECC_i = \sum_{j=0}^i VCEC_j$ by construction, we compute in time $O(m)$ with basic dynamic programming – although since m is small, a $\log(m)$ parallel algorithm is a practical alternative on GPUs [17].

Faces introduced by a voxel. We say that a face is *introduced by a voxel*, if this voxel has the smallest value among all voxels containing the given face (in other words: if the face inherits the value from this voxel). One caveat is that ties have to be broken in a consistent manner: if two voxels have the same value then we prefer the one with a lexicographically lower position in the input array. Later we show that this turns in a fast computation, without the need to explicitly compare the indices.

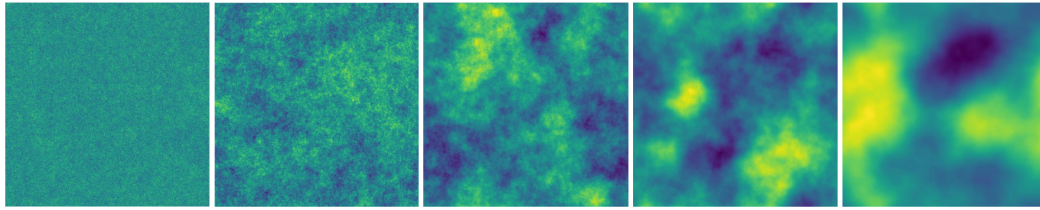


■ **Figure 1 (bottom)** Input image at three thresholds. Their grayscale values correspond to terrain elevations. **(top)** The three plots share the x-axis which represents the thresholds of the input image. The topmost plot shows the persistence diagram (rotated for clarity). For each threshold, it marks the lifetime of topological features: connected components in red, and holes in blue. The three highlighted areas show the features alive at the three corresponding thresholds, which are visualized as the Betti curves below. The ECC is the pointwise difference between the curves. This example highlights the main downside of ECC: its reliance on counts of topological features, while persistence also distinguishes their prominence.

Sequential algorithm. With this we can sketch a simple sequential algorithm for the computation of VCEC for an image (see Algorithm 1). Note that there is no need to explicitly store any information of the lower dimensional cells. This algorithm will be a basis for our GPU algorithm.

3 Related work on ECC: applications and computations

Due to their simplicity, both the Euler characteristic curve (ECC) and the Euler characteristic (EC) find usage in many fields – especially the ones related to imaging. A great introduction to the topic is the review paper by Worsley [23] from which we sample some of the applications below. We then discuss the related work on the computational side.



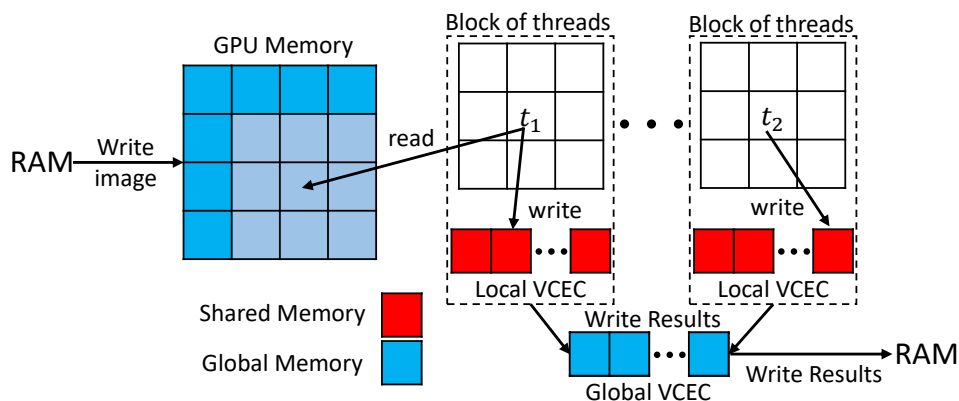
■ **Figure 2** Visualizations of Gaussian random fields generated with different levels of smoothness.

Applications. Ideas related to the EC were present in astrophysics already in 1970s (EC is called the *genus curve*). They were formalized in 1986 by Gott and others [7] in the study of the sponge-like topology of the large-scale structures in the universe; later ECC became an important tool in the study of the imaging data describing the cosmic microwave background (CMB) radiation [14]. This is closely related to earlier work on the topology of Gaussian random fields (GRFs) by Adler and Hasofer [1] – GRFs are used to model the CMB. See Fig. 2 for images of GRFs. Ideas related to the EC were popular in the field of bone morphometry. They were formalized mathematically in 1993 [15]; there EC was used to characterize the trabecular structures in bones – particularly to compute the first Betti number (called the *connectivity* in this field). EC is a common tool in *morphological image processing* [9]; it is widely used to characterize the shape of thresholded (binary) images under the name *Euler number*; later it was computed at all thresholds of a grayscale image – this is ECC hiding under the name *stable Euler number* [18]. In particular the zero-crossing of the ECC is used to select a segmentation threshold; see Fig. 1 for a rudimentary example showing that the riverbeds in a terrain are clearly highlighted at this threshold.

ECC in TDA. Many of the above applications are close to the way topological descriptors are used today in topological data analysis (TDA), although persistent homology is a much more popular choice. Still, there is a number of recent TDA studies using EC and ECC. Bobrowski and Skraba [4] demonstrate that ECC is surprisingly powerful in analyzing the percolation threshold in random cubical filtrations (and other random models). Crawford and collaborators propose [5] as a novel statistic based on EC; it proves useful in predicting clinical outcomes of brain cancer based on brain imaging data. Amezcua and collaborators [2] analyzed the shape of barley using an image transform based on EC.

Computations. However, the employed algorithmic techniques were different from the simple setup outlined in the previous section. Instead, the computations often exploited the connection of EC with differential geometry. In particular, an explicit, efficient algorithm for EC of 3D voxel data was presented in [7]. Its efficiency stems from precomputed tables of voxel neighbourhood. Our approach is different and based on the mathematical-algorithmic setup of cubical homology. This direction emerged in the 1990s in the work of Kaczynski, Mischaikow and Mrozek. Originating in the context of computational dynamics, it evolved in a more general framework described in their book [11]. The first efficient, general-dimension algorithm for EC of binary images uses this setup. The algorithm is due to Ziou and Allili [26] in 2001. The idea is to view a binary image as a cubical complex and compactly encode this information. Compared to existing algorithms, this approach is simple, efficient, and it works in arbitrary dimension. One drawback is the memory overhead related to storing the cubical complex.

The first efficient algorithm for ECC computation is presented by Snidaro and Foresti [18] in 2003. It focused on 2D images. The first efficient algorithm for ECC for 3D images – which also works in arbitrary dimension – is due to Heiss and Wagner [8] in 2017. This approach extends the idea of Ziou and Allili to cubical filtrations (i.e. from binary to grayscale data). It also offers improvements: the cubical complex is not stored explicitly; computations are done in parallel; the image is streamed into memory in small chunks so that images of arbitrary size can be handled. Our GPU implementation is based on this approach.



■ **Figure 3** The image is first copied from RAM to GPU’s global memory. Each block of threads is responsible for a patch of the image. Each thread in the block needs to access a voxel and its eight neighbours (all marked in lighter blue). When the block is done, the block’s local result is added to the global result. The final result for the entire image is transferred to RAM.

4 GPU implementation

Our GPU implementation is illustrated in Algorithm 2. The listed code is slightly simplified for readability and covers only the case of 2-dimensional images with grayscale levels ranging from 0 to 255. After we cover the overall structure of computations, we explain the implementation in details.

4.1 Challenges

Our GPU implementation tackles three main challenges: (1) We needed to adapt the CPU algorithm to the GPU setting to fully exploit its massive parallelism. The main challenge was that instead a dozen of threads we had to manage thousands of threads working in parallel – which required us to structure the computations differently. Further, unlike the CPU version, which employed a simple lock-free scheme, we needed to explicitly deal with race conditions and other issues related to synchronization. Apart from ensuring correctness, we had to experiment with different synchronization granularities to achieve optimized performance.

(2) Efficient use of GPU’s memory hierarchy and its limited resources. GPU is notorious for its complicated memory hierarchy and limited memory resources. Unlike CPU programming these have to be explicitly incorporated into algorithmic design. We managed to craft an efficient multi-level caching hierarchy, taking into account the access patterns characteristic of working with cubical complexes. With careful analysis of access probabilities, we managed to ensure that only a single voxel (and not 9 or 27) per thread is fetched from main memory.

(3) Many of the technicalities are not visible when analyzing the GPU kernel. One particular

technical difficulty was achieving streaming operation without affecting the performance. This enabled us to handle inputs of virtually unlimited size, despite limited GPU memory. We also organized the streaming processing in a pipeline which allows to overlap the computations with memory transfers.

4.2 Structure of the computations

The C++ implementation shown in Algorithm 2 defines a *compute kernel* which is the computation realized by a thread. Thread-centric view is assumed hereafter, and we will talk about *the thread* remembering that the same computations are done concurrently by many threads.

Single thread. Each thread handles a single voxel, namely, realizes lines 3–5 of Algorithm 1. In other words, each thread iterates over the faces of a given voxel, decides which of them are introduced by this voxel, and updates the VCEC vector at the value of the voxel.

Blocks of threads. Threads are grouped into *blocks*, and we can imagine that the image is decomposed into rectangular patches. Many such patches are processed concurrently – although not necessarily in parallel. See Fig. 3 for an overview.

4.3 Optimizations

Computations structured this way perfectly fit the GPU pipeline – however, using the potential of the hardware requires careful memory management. Specifically, GPUs have a hierarchy of memory types with different sizes and performance characteristics. Unlike CPU programming, the programmer must make thoughtful use of these various kinds of memories.

Below we explain some of our implementation choices, and mention common pitfalls. To illustrate these issues, we start from a hypothetical naive implementation directly implementing Algorithm 1 in GPU. We then improve it step by step, arriving at the implementation listed in Algorithm 2.

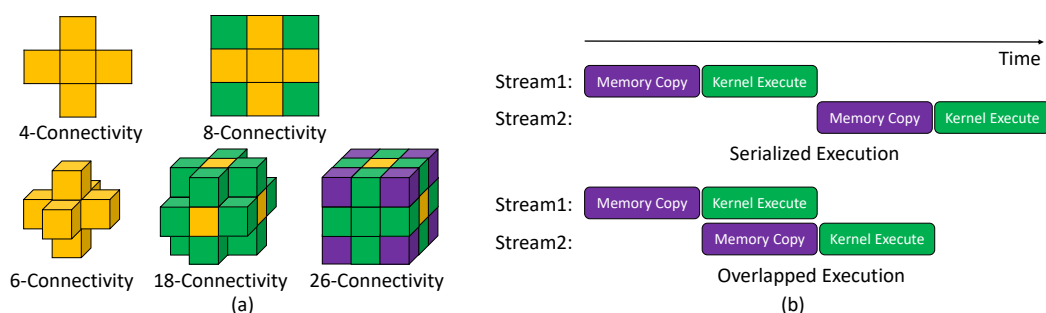
Location of input image. The image data is initially copied from main memory (RAM) to the GPU's *global memory* – this is the only type of GPU memory large enough to store an image of a reasonable size. Note that large images may exceed the size of the global memory. For now we assume that the image fits in global memory – we solve this issue in Section 4.4.

Race conditions. We store the VCEC in global memory, simply as an array of 256 integers. However, since multiple threads will update the same memory location, we need to be wary of *race conditions*. Modern GPUs offer efficient implementation of *atomic operations*, including *atomicAdd*, which ensures that all the updates to VCEC will be correctly recorded. However, all updates issued simultaneously on a single memory location will be *serialized* – which means we lose the main advantage of the GPU hardware, namely, massive parallelism.

Using registers. We can mitigate the above problem by accumulating the contribution of the given voxel in a *register*. Registers provide the fastest type of GPU memory. Additionally, they are local to each thread, which means that we do not need to worry about race conditions when updating the values stored in them. We still need to update the global VCEC using *atomicAdd*, but the number of updates is now one per thread (instead of 9 in 2D or 27 in 3D).

Shared memory. The above is an improvement but still far from ideal. The next step is to use *shared memory*. This is another type of low-latency memory offered by GPUs, although slower than registers. It is shared between all threads in a given block. So instead of updating the global VCEC, each thread updates the local VCEC of its own block. This local VCEC is simply an array of 256 integers, as declared in line 6.

We still need to use `atomicAdd` to avoid race conditions – but now the collision probability is lower, since only threads belonging to a single block can access a given shared memory location. And since the size of the block is configurable, we can find the size which yields good performance. This optimization has two additional advantages: shared memory has significantly lower latency than global memory; in modern GPUs atomic operations on shared memory are significantly more efficient than on global memory. This is not true on older GPUs, and using them would require a more elaborate way of merging the results.



■ **Figure 4** (a) Various kinds of connectivity relations between voxels. (b) An illustration showing that overlapping the computations and memory transfer can reduce the overall execution time.

Parallel initialization and finalization. The shared memory needs to be initialized, and its final content needs to be added to the global VCEC vector. We perform both steps in parallel: a single thread in the block is responsible for one location in the shared-memory array. In line 10 the thread sets a specific location to zero – or does nothing. Similarly, in line 32 the thread issues an `atomicAdd`, which updates the global result with the local one. Note that in these two cases the index does not depend on the value of the voxel assigned to the thread – we simply compute the unique number of the thread within its block; see line 7. We use it to index the shared array.

Block-level synchronization. Since the threads in the block are not guaranteed to run in parallel, we need to synchronize them – otherwise they could start working on uninitialized memory, or update the global result using unfinished local results. Proper synchronization is insured by placing a *block-level synchronizing barrier* in lines 11 and 33.

Accessing neighbors. To decide which cells are introduced by a given voxel, we compare the value of the voxel with its neighbours (with careful tie-breaking). Specifically, we access the 8-connectivity neighbors of a given voxel (and 26 in 3D); see Fig. 4(a) for an illustration.

Texture cache. Accessing these values is another source of inefficiency, linked with the high latency of the global memory. We mitigate this by using a specialized caching mechanism, called *texture cache*. It is often referred to as *texture memory*, which is misleading since modern GPUs realize this as a caching layer on top of data residing in global memory. When

the value of a voxel is requested from global memory using this mechanism, the neighbours of the accessed voxel are automatically cached in GPU's specialized low-latency memory. It is as fast as shared memory. Line 17 shows how a voxel value is requested via the texture cache. This is a read-only cache, which suits our algorithm well, since we are not modifying the image.

The texture cache is optimized exactly for the *spatial access locality* displayed by ECC computations. Also, since neighboring voxels are generally processed in parallel, it is likely that the neighbors' values already reside in the fast cache. Overall, we can expect that on average only a single uncached global memory access will be required per thread – but there is no guarantee due to the limited size of the cache.

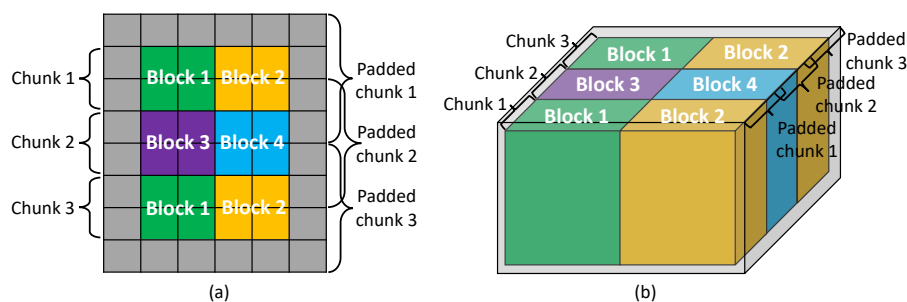
The danger of register spilling. We store the frequently used voxel values in *registers*. In the case of 2D inputs, the 4-connectivity neighbors (marked in yellow in Fig. 4(a)) are involved in 3 different comparison operations and therefore we cache them in registers. The 8-connectivity voxels (marked in green) are read once and used once, so we save registers and rely on the aforementioned texture cache. Similarly for 3D inputs, only the 6-connectivity voxels (used 9 times) and 18-connectivity voxels (used 3 times) are stored in registers. To use the registers, we unroll the loop, namely, replace it with a series of statements.

■ **Algorithm 2** Implementation of the VCEC on GPU for a 2D image.

```

__constant__ int image_width, image_height;
const int num_bins = 256;
1
2
3
__global__ void vcec_kernel(cudaTextureObject_t voxels, int* vcec_global)
4
{
5
    __shared__ int vcec_local[num_bins];
6
    const int thread_number = blockDim.x * threadIdx.y + threadIdx.x;
7
8
    if (thread_number < num_bins)
9
        vcec_local[thread_number] = 0;
10
    __syncthreads();
11
12
    const int ix = blockDim.x * blockIdx.x + threadIdx.x + 1;
13
    const int iy = blockDim.y * blockIdx.y + threadIdx.y + 1;
14
    if (ix >= image_width + 1 || iy >= image_height + 1) return;
15
16
    int change = 1;
17
    int c = tex2D<float>(voxels, ix, iy);
18
    int t = tex2D<float>(voxels, ix, iy - 1);
19
    int b = tex2D<float>(voxels, ix, iy + 1);
20
    int l = tex2D<float>(voxels, ix - 1, iy);
21
    int r = tex2D<float>(voxels, ix + 1, iy);
22
23
    // Vertices
24
    change+=(c < l && c < t && c < tex2D<float>(voxels, ix - 1, iy - 1));
25
    change+=(c < t && c <= r && c < tex2D<float>(voxels, ix + 1, iy - 1));
26
    change+=(c < l && c <= b && c <= tex2D<float>(voxels, ix - 1, iy + 1));
27
    change+=(c <= b && c <= r && c <= tex2D<float>(voxels, ix + 1, iy + 1));
28
29
    // Edges
30
    change -= ((c < t) + (c < l) + (c <= r) + (c <= b));
31
32
    atomicAdd(&vcec_local[c], change);
33
    __syncthreads();
34
    if (thread_number < num_bins)
35
        atomicAdd(&vcec_global[thread_number], vcec_local[thread_number]);
36
}

```



■ **Figure 5** Chunking in 2D and 3D. We emphasize that the chunks are a disjoint decomposition of the input – but the padded chunks are not. This extra padding provides information necessary to ensure that each cell introduced by input voxels is counted exactly once.

It may seem like a good idea to cache everything in registers. This is especially misleading since defining register variables is syntactically the same as defining stack-allocated variables in C++ CPU programming; see line 17 for an example. We are careful with register allocation – one common pitfall is *register spilling*. One danger stems from the fact that the number of registers per block is limited by hardware – but the number of threads in the block is configurable. So requesting a block of a certain size may cause the number of required blocks to exceed the availability. In this case the values are – silently! – stored in what is called *local memory* – which is a misnomer because the values are physically placed in global memory. So instead of using the fastest memory, the slowest one is used. This simple mistake can cause a performance hit of two orders of magnitude.

Branching. GPU performance can be significantly penalized by branching and loops. In general, GPU programs allow for considerable flexibility – but they operate most efficiently as SIMD (single instruction multiple data) units. In other words, kernels whose flow of execution does not depend on the input data are preferred.

Warps of threads. The above is related to how the threads are scheduled by GPU. Namely, the threads within each block are additionally grouped into *warps* of 32 threads. Any branch (i.e. if statement) splits the execution of the entire warp into two divergent paths. This is often called *intra-warp branching*. These two paths generally cannot be executed in parallel, instead they are serialized. So a single thread can cause all the remaining threads in its warp to remain idle, limiting parallel execution. We avoid branching in several ways: we surround the data with a collar of voxels with infinite value, to avoid divergent branches due to boundary conditions. We also update the *change* variable without a branching statement, see e.g. line 25 where we add the truth value of a logical expression, even if it evaluates to 0.

Constant memory. Kernels often require additional information, e.g. the width and height of the processed image. Accessing such information from global memory multiple times would be inefficient. Instead, we ensure quick access by declaring such variables as constant memory, see e.g. line 1. With this, the values are stored in GPU's *constant memory*, which is specialized for fast *broadcast* of stored values to multiple threads.

4.4 Streaming

In practice, a lot of 3D images are too large to fit in GPU memory. We overcome this obstacle by dividing an input into *chunks* and process them separately. Another benefit of streaming lies in the CUDA's asynchronous behaviour which allows us to overlap data transfers and kernel executions. Breaking an input into smaller pieces helps hide the high latency related to memory transfers between GPU and RAM; see Fig. 4.

Input of size (w_0, w_1, \dots) is cut along the first coordinate. This way the image is divided into c *chunks* of size at most $(\lfloor \frac{w_0}{c} \rfloor, w_1, \dots)$. This ensures that the resulting chunks correspond to contiguous memory addresses as arrays are stored in row-major order in C++. As illustrated in Fig. 5, we extend the chunks by a single-voxel padding. The collar contains either the value of a voxel – to ensure that each input voxel has access to all of its neighbours; or positive infinity – as explained before.

Each chunk is loaded into the GPU memory (including the collar). The chunk is then processed with one or multiple CUDA blocks depending on the size. After finishing computations, the free blocks will be reassigned to a new chunk for computation.

Overlapping computations and data transfers. CUDA devices contain engines for various tasks. Modern devices typically have two copy engines, one for host-to-device transfers and another for device-to-host transfers, as well as a kernel engine. With pinned (non-pageable) host memory, the tasks launched into non-default different CUDA streams can be executed concurrently assuming no dependencies amongst them. In other words, loading a chunk into device, writing results back to host, and kernel execution can happen simultaneously. With a reasonable choice of c , the overhead of data transfer can be greatly alleviated. Fig. 4(b) illustrates a simplified case of overlapping transfers and kernel executions. Suppose we have equal running time for memory copy and kernel execution. Compared to serialized execution, overlapped execution practically hides the kernel execution time for one chunk when $c = 2$.

5 Experiments

We use the C++ compiler shipped with Visual Studio 2019 (v142) and language standard of C++14 for the compilations of both CPU and GPU implementations. The following experiments are conducted on a desktop machine with Intel Core i7-9700K CPU with 8 physical cores (and disabled hyper-threading), 16GB of RAM, Sabrent Rocket Q 2TB NVMe PCIe M.2 2280 SSD drive, and a NVIDIA RTX 2070 graphics card with 8GB of GDDR6 memory. It is a modern commodity workstation.

Datasets. We use a mix of synthetic and real-world datasets:

- Cosmic microwave background (CMB) imaging data comes from astrophysical measurements. The original data is on a 2-dimensional sphere; we use a single image projection in different resolution. Each image contains at most 256 unique values.
- Virtual Imaging Clinical Trials for Regulatory Evaluation (VICTRE) [3] project provides realistic simulation of breast phantoms. We generated 20 3D breast volumes. Each image contains only 11 unique values.
- We also use a set of 70 2D Gaussian Random Fields (GRF) with 7 sizes (10 samples for each size) and 30 3D GRFs with 3 sizes (10 samples each). Each image contains only 1024 unique values.
- For larger experiments we use data generated by sampling the uniform distribution for each voxel. We call this data uniform noise.

■ **Table 1** This table compares the execution time of the CPU and GPU implementations. These are end-to-end timings, include disk I/O and the GPU overhead related to initializing our computations. The two rightmost columns are relevant in situations in which the input resides in GPU memory.

	Input size(B)	CPU overall	GPU overall	Overall speedup	CPU disk read	GPU disk read	GPU over-head	GPU exec. (kernel)	GPU kernel Gvox/s	
Uniform Noise										
	4096 ³	256G	37.72m	9.10m	4.14x	7.30m	9.08m	0.67s	0.20m	5.62
	2048 ³	32G	4.86m	0.71m	6.77x	0.99m	0.71m	0.41s	0.03m	5.61
	1024 ³	4G	36.85s	5.63s	6.55x	6.85s	5.20s	0.37s	0.16s	6.57
	512 ³	512M	4.97s	0.85s	5.86x	1.00s	0.64s	0.19s	0.02s	6.55
Gaussian Random Field										
	512 ³	512M	4.93s	0.86s	5.75x	0.90s	0.66s	0.19s	20.88ms	6.43
	256 ³	64M	0.63s	0.24s	2.58x	0.13s	0.09s	0.15s	2.64ms	6.35
	128 ³	8M	0.11s	0.12s	0.86x	0.02s	0.01s	0.12s	0.35ms	6.03
	8192 ³	256M	1.47s	0.53s	2.75x	0.44s	0.36s	0.16s	6.64ms	10.10
	4096 ³	64M	0.38s	0.21s	1.84x	0.12s	0.08s	0.14s	1.74ms	9.67
	2048 ³	16M	0.09s	0.18s	0.55x	0.04s	0.03s	0.12s	0.45ms	9.36
VICTRE										
	287 359 202	79.3M	0.59s	0.30s	1.98x	0.16s	0.13s	0.14s	3.85ms	5.41
	440 518 488	424M	2.99s	0.77s	3.87x	0.98s	0.45s	0.24s	20.65ms	5.39
	434 446 384	147M	1.11s	0.36s	3.02x	0.29s	0.15s	0.16s	7.13ms	5.40
	434 446 384	283M	1.96s	0.53s	3.70x	0.79s	0.30s	0.18s	13.72ms	5.42
CMB										
	1500 750	1.07M	0.03s	0.12s	0.22x	0.01s	0.01s	0.11s	0.15ms	7.40
	3000 1500	4.29M	0.09s	0.15s	0.61x	0.04s	0.02s	0.13s	0.44ms	10.16
	6400 3200	19.5M	0.37s	0.25s	1.49x	0.13s	0.08s	0.14s	1.94ms	10.56

All datasets except for CMB are stored in binary format as 32 bit IEEE 754 floating point values. CMB is stored in binary format as 8-bit unsigned integer values.

Voxel throughput. We are mostly interested in the size (counted in numbers of pixels or voxels) of the image that can be processed in a second. We call this quantity the *voxel throughput* and express it in GVox/s, namely billions (10^9) voxels per second. All time measurements are given in ms (milliseconds, 10^{-3} s).

5.1 Case study: Single image on disk

In this case we employ CHUNKEYEuler by Heiss and Wagner [8] as a CPU baseline. CHUNKEYEuler is the state-of-the-art CPU parallel streaming ECC implementation. To the best of our knowledge, no other software can handle the sizes of the data we experiment with. We run experiments with all eight available CPU cores.

Overall execution time. In this setup, we simply measure the overall execution time including reading the image from disk; see Table 1. For files smaller than around 16MB, the CPU version is faster. This is due to the overhead related to initializing our GPU computations. For files larger than 0.5GB, the GPU version is between 4 to 6 times faster – although it is severely limited by disk I/O which takes between 75% and 99.7% of its total execution time.

Streaming. Note that we handle files significantly larger than the available 8GB GPU memory and 16GB RAM. This is achieved by a streaming algorithm described before. This was a major difficulty and is described in Section 5. In particular, we handled an image of size 4096³ which takes 0.25TB.

■ **Table 2** This table shows the timings for the pipeline involving the iterated ECC and Gaussian smoothing computations. The key observation is that when averaged over multiple iterations the overall time is dominated by the two kernel executions. This confirms that there are no additional bottlenecks in this pipeline, and especially in our ECC computations. Note that the image is read once, and so the time to load the image from disk is a one-time cost.

	Overall	Overall	ECC mem.	ECC exec.	Gaussian	Disk
	Overall	avg.	avg.	avg.	exec. avg.	read
	[ms]	[ms]	[ms]	[ms]	[ms]	[ms]
Uniform Noise						
(ECC+Gaussian) \times 1	137.16	137.16	0.28	0.16	1.55	7.72
(ECC+Gaussian) \times 10	172.80	17.28	0.06	0.15	0.20	7.38
(ECC+Gaussian) \times 100	149.96	1.50	0.03	0.13	0.09	7.81
(ECC+Gaussian) \times 1000	352.02	0.35	0.03	0.12	0.07	7.22
(ECC+Gaussian) \times 1000	2786.64	0.28	0.03	0.17	0.07	7.57

GPU overhead. The overhead mentioned above is related to the initialization and shutdown of the GPU device, and memory allocation specific to our implementation. This overhead ranges between 100 and 700ms and is a one-time cost. This is why GPU is more effective for larger datasets – but also for batches of smaller ones. We will focus on that next.

5.2 Case study: Batch processing of images on disk

In this case we read multiple files from disk. We focus on small files, because they were problematic for the GPU implementation (due to the GPU overhead). Table 3 shows that the overhead now amortizes when many files are processed. This means that in batch processing the GPU implementation is always preferred over the CPU one. Still, this is not an ideal setup for GPU, since the computations are heavily limited by disk I/O.

Prospects. The above issue opens up a new avenue – it may now be opportune to load compressed images, which would limit the disk I/O time. We plan to investigate this in future work.

5.3 Case study: GPU-only pipeline

In this scenario, the images are stored and processed entirely in GPU memory. This emulates pipelines implemented entirely on GPUs, such as some implementations of CNNs [13]. As mentioned earlier, this case is our primary motivation. We are trying to determine if our ECC kernel could be part of such a GPU pipeline without becoming a significant performance bottleneck. We also need to verify that our computational setup does not incur any unexpected additional bottlenecks.

Pipeline. To this aim, we consider a two-step pipeline: (1) compute the ECC; (2) apply a Gaussian smoothing filter. Steps (1) and (2) are performed repeatedly on an image stored in GPU memory. We iterate up to 10000 times using a 1024^2 GRF image. After each iteration, the resulting VCEC is transferred to RAM and post-processed, including computing ECC.

Gaussian smoothing implementation. We implement the Gaussian smoothing filter as a discrete Gaussian convolution. We exploit its separability and use a highly optimized GPU kernel. We use a Gaussian kernel width of 13 pixels (see also Fig. 6).

■ **Table 3** We show timings averaged over running different numbers of files. This table confirms that the GPU overhead, which dominates the computations for a single small file, amortizes across many samples. It is clear that the GPU performance is heavily limited by disk I/O.

	Input size(B)	GPU overall avg. [ms]	GPU disk read avg. [ms]
Uniform Noise			
$128^2 \times 1$	64K	119.83	0.69
$128^2 \times 100$	6.25M	1.77	0.46
$128^2 \times 1000$	62.5M	0.66	0.45
$128^2 \times 10000$	625M	0.52	0.42
Gaussian Random Field			
$128^3 \times 1$	8M	124.68	12.02
$128^3 \times 10$	80M	28.13	13.86
$128^3 \times 100$	800M	15.38	13.82
$128^3 \times 1000$	8000M	11.96	11.67



■ **Figure 6** Images at consecutive steps in the smoothing pipeline.

Potential performance bottlenecks. Since the initial image is loaded into GPU memory once, the cost of reading from disk amortizes across many kernel runs. As we already checked, the same applies to the GPU overhead. Column “ECC mem” in Table 2 shows the cost of transferring the resulting VCEC from GPU memory to RAM and the cost of its CPU post-processing; this does not incur a performance hit either. Overall, we see that the kernel executions dominate the overall time.

Performance comparison. We can therefore directly compare the performance of the ECC kernel and the convolution kernel. Table 2 shows that the throughput of the two kernels is at the same order of magnitude. The Gaussian kernel is up to 2.5 times faster. However, the impact on the overall performance of a CNN is likely to be significantly lower, since a single convolution often contributes less than half of the total computation time performed by a convolution layer in a CNN [13].

ECC kernel performance. We highlight the performance of the ECC kernel. The throughput is between 5 and 10 GVox/s. To put things in perspective, it allows us to handle:

1. 3D images of size 512^3 voxels at the rate of 30Hz;
2. 2D images of 8K resolution (7680×4320 pixels) at the rate of 120Hz.

5.4 Dependence on dimension

Perhaps surprisingly, the performance does not depend on the dimension of the image – which suggests that the caching hierarchy we devised works well – and that the neighbours

are typically retrieved from the cache. This way the dependence on the number of neighbours (8 vs 26) largely disappears. This property would not extend to higher dimensions, since the texture cache is only available in dimensions that are smaller or equal to 3.

6 Discussion

We proposed an efficient GPU implementation to compute the Euler characteristic curve of imaging data. The resulting software is highly practical. Its three major advantages are:

- High speed: for images present in GPU memory, it processes images at speed exceeding 5×10^9 voxels per second. This is a realistic scenario for example in the context of convolutional networks.
- Streaming: it can handle images of virtually unlimited size. This is crucial since GPU memory is a limited resource.
- ECC contains topological information which was successfully used in many application domains.

We believe these results open up interesting avenues. Our plans are twofold. First, we intend to integrate our ECC computations into CNNs. With the efficiency gap closed, we hope that topological methods will start permeating mainstream machine learning. Second, we hope that the full power of persistent homology can be used in such contexts. With the gathered experience specific to handling cubical filtrations on GPUs, we hope to make the first steps towards designing GPU algorithms for persistence analysis of imaging data.

References


- 1 Robert J. Adler and A. M. Hasofer. Level Crossings for Random Fields. *The Annals of Probability*, 4(1):1–12, 1976. doi:10.1214/aop/1176996176.
- 2 Erik J Amezcuita, Michelle Quigley, Tim Ophelders, Jacob Landis, Elizabeth Munch, Daniel Chitwood, and Daniel Koenig. Quantifying barley morphology using the Euler characteristic transform. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020.
- 3 Aldo Badano, Christian G. Graff, Andreu Badal, Diksha Sharma, Rongping Zeng, Frank W. Samuelson, Stephen J. Glick, and Kyle J. Myers. Evaluation of Digital Breast Tomosynthesis as Replacement of Full-Field Digital Mammography Using an In Silico Imaging Trial. *JAMA Network Open*, 1(7):e185474–e185474, November 2018. doi:10.1001/jamanetworkopen.2018.5474.
- 4 Omer Bobrowski and Primoz Skraba. Homological percolation and the Euler characteristic. *Phys. Rev. E*, 101:032304, March 2020. doi:10.1103/PhysRevE.101.032304.
- 5 Lorin Crawford, Anthea Monod, Andrew X Chen, Sayan Mukherjee, and Raúl Rabadán. Predicting clinical outcomes in glioblastoma: an application of topological and functional data analysis. *Journal of the American Statistical Association*, 115(531):1139–1150, 2020.
- 6 Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.
- 7 J Richard Gott III, Adrian L Melott, and Mark Dickinson. The sponge-like topology of large-scale structure in the universe. *The Astrophysical Journal*, 306:341–357, 1986.
- 8 Teresa Heiss and Hubert Wagner. Streaming algorithm for Euler characteristic curves of multidimensional images. In Michael Felsberg, Anders Heyden, and Norbert Krüger, editors, *Computer Analysis of Images and Patterns - 17th International Conference, CAIP 2017, Ystad, Sweden, August 22-24, 2017, Proceedings, Part I*, volume 10424 of *Lecture Notes in Computer Science*, pages 397–409. Springer, 2017. doi:10.1007/978-3-319-64689-3_32.
- 9 Berthold Horn, Berthold Klaus, and Paul Horn. *Robot vision*. The MIT Press, 1986.

- 10 Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- 11 Tomasz Kaczynski, Konstantin Mischaikow, and Marion Mrozek. Computational homology. *Bull. Amer. Math. Soc.*, 43:255–258, 2006.
- 12 Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- 13 Xiaqing Li, Guangyan Zhang, H Howie Huang, Zhufan Wang, and Weimin Zheng. Performance analysis of GPU-based convolutional neural networks. In *2016 45th International conference on parallel processing (ICPP)*, pages 67–76. IEEE, 2016.
- 14 Dmitri I. Novikov, Hume A. Feldman, and Sergei F. Shandarin. Minkowski functionals and cluster analysis for CMB maps. *International Journal of Modern Physics D*, 08(03):291–306, 1999. doi:10.1142/S0218271899000225.
- 15 A Odgaard and HJG Gundersen. Quantification of connectivity in cancellous bone, with special emphasis on 3-D reconstructions. *Bone*, 14(2):173–182, 1993.
- 16 Johanna Pasquet, Emmanuel Bertin, Marie Treyer, Stéphane Arnouts, and Dominique Fouchez. Photometric redshifts from SDSS images using a convolutional neural network. *Astronomy & Astrophysics*, 621:A26, 2019.
- 17 Shubhabrata Sengupta, Mark Harris, Michael Garland, et al. Efficient parallel scan algorithms for GPUs. *NVIDIA, Santa Clara, CA, Tech. Rep. NVR-2008-003*, 1(1):1–17, 2008.
- 18 L. Snidaro and G. L. Foresti. Real-time thresholding with Euler numbers. *Pattern Recogn. Lett.*, 24(9–10):1533–1544, June 2003. doi:10.1016/S0167-8655(02)00392-6.
- 19 Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.
- 20 Hubert Wagner, Chao Chen, and Erald Vuçini. Efficient computation of persistent homology for cubical data. In *Topological methods in data analysis and visualization II*, pages 91–106. Springer, 2012.
- 21 Fan Wang, Saarthak Kapse, Steven Liu, Prateek Prasanna, and Chao Chen. TopoTxR: A Topological Biomarker for Predicting Treatment Response in Breast Cancer. In *Information Processing in Medical Imaging - 27th International Conference, IPMI*, volume 12729 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 2021. doi:10.1007/978-3-030-78191-0_30.
- 22 Fan Wang, Huidong Liu, Dimitris Samaras, and Chao Chen. TopoGAN: A Topology-Aware Generative Adversarial Network. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III*, pages 118–136. Springer-Verlag, 2020. doi:10.1007/978-3-030-58580-8_8.
- 23 Keith J Worsley. The geometry of random images. *Chance*, 9(1):27–40, 1996.
- 24 Pengxiang Wu, Chao Chen, Yusu Wang, Shaoting Zhang, Changhe Yuan, Zhen Qian, Dimitris Metaxas, and Leon Axel. Optimal Topological Cycles and Their Application in Cardiac Trabeculae Restoration. In *In International Conference on Information Processing in Medical Imaging (IPMI), 2017*, pages 80–92, May 2017. doi:10.1007/978-3-319-59050-9_7.
- 25 Simon Zhang, Mengbai Xiao, and Hao Wang. GPU-Accelerated Computation of Vietoris-Rips Persistence Barcodes. In *36th International Symposium on Computational Geometry (SoCG 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- 26 Djemel Ziou and Madjid Allili. Generating cubical complexes from image data and computation of the Euler number. *Pattern Recognition*, 35(12):2833–2839, 2002. Pattern Recognition in Information Systems. doi:10.1016/S0031-3203(01)00238-2.

Space Ants: Episode II – Coordinating Connected Catoms

Julien Bourgeois ✉ 


FEMTO-ST Institute, University of Bourgogne Franche-Comté, CNRS, Montbeliard, France

Sándor P. Fekete ✉ 

Department of Computer Science, TU Braunschweig, Germany

Ramin Kosfeld ✉ 

Department of Computer Science, TU Braunschweig, Germany

Peter Kramer ✉ 

Department of Computer Science, TU Braunschweig, Germany

Benoît Piranda ✉ 

FEMTO-ST Institute, University of Bourgogne Franche-Comté, CNRS, Montbeliard, France

Christian Rieck ✉ 

Department of Computer Science, TU Braunschweig, Germany

Christian Scheffer ✉ 

Faculty of Electrical Engineering and Computer Science, Hochschule Bochum, Germany

Abstract

How can a set of identical mobile agents coordinate their motions to transform their arrangement from a given starting to a desired goal configuration? We consider this question in the context of actual physical devices called *Catoms*, which can perform reconfiguration, but need to maintain connectivity at all times to ensure communication and energy supply. We demonstrate and animate algorithmic results, in particular a proof of hardness, as well as an algorithm that guarantees *constant stretch* for certain classes of arrangements: If mapping the start configuration to the target configuration requires a maximum Manhattan distance of d , then the total duration of our overall schedule is in $\mathcal{O}(d)$, which is optimal up to constant factors.

2012 ACM Subject Classification Theory of computation → Computational geometry; Computing methodologies → Motion path planning

Keywords and phrases Motion planning, parallel motion, bounded stretch, scaled shape, makespan, connectivity, swarm robotics

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.65

Category Media Exposition

1 Introduction

Coordinating the motion of a set of objects is a fundamental problem that occurs in a large spectrum of theoretical contexts and practical applications. A typical task arises from relocating a large collection of agents from a given start into a desired target configuration, while avoiding collisions between objects or with obstacles.

A crucial algorithmic aspect is *efficiency*: How can we reach the target configuration in a timely or energy-efficient manner? Exploiting parallelism in a robot swarm to achieve an efficient schedule was studied by Demaine et al. [2, 4], who showed that under certain conditions, a labeled set of robots can be reconfigured with bounded *stretch*, i.e., there is a collision-free motion plan such that the overall length of the schedule (the *makespan*) remains within a constant of the lower bound that arises from the maximum distance between origin



© Julien Bourgeois, Sándor P. Fekete, Ramin Kosfeld, Peter Kramer, Benoît Piranda, Christian Rieck, and Christian Scheffer;

licensed under Creative Commons License CC-BY 4.0

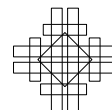
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 65; pp. 65:1–65:6

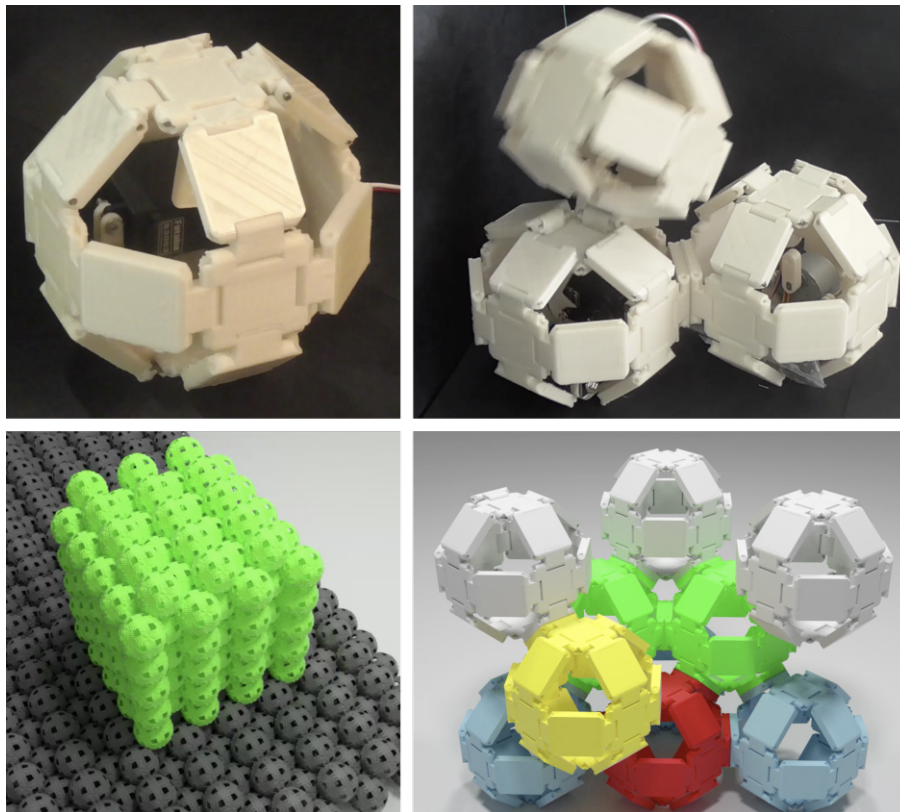
Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



and destination of individual robots. Practical computation of minimum makespan schedules for a set of benchmark instances was also the subject of the 2021 Computational Geometry Challenge; see [6] for an overview, and [3, 8, 13] for successful contributions.



■ **Figure 1** (Top left) A Datom. (Top right) A Datom performing a local move between neighbors. (Bottom right) A local arrangement. (Bottom left) A large-scale arrangement of 3D Catoms.

A practical application arises from coordinating a set of *Datoms* (for “Deformable Atom” as a reference to the Claytronics Atom, *Catom* [7]), which are small-scale electronic devices that can change their shape and interact with their neighbors to allow communication, energy supply, and rearrangement; see [9, 10, 12]. This requires maintaining *connectivity* of the overall arrangement, which is not guaranteed by the approach of Demaine et al. [4].

In this contribution, we illustrate and animate recent algorithmic results by Fekete et al. [5], who presented an approach that does achieve constant stretch for *connected, unlabeled* swarms of robots for the class of *scaled* arrangements; such arrangements arise by increasing all dimensions of a given object by the same multiplicative factor and have been considered in previous seminal work on self-assembly, often with unbounded or logarithmic scale factors (along the lines of what has been considered in self-assembly [11]). The method by Fekete et al. [5] relies strongly on the exchangeability of indistinguishable robots, which allows a high flexibility in allocating robots to target destinations.

This also adds to previous work [1] on efficient reconfiguration of large-scale arrangements. *Space Ants: Episode I – The Rise of the Machines* considers recognition and reconfiguration of lattice-based cellular structures by very simple robots with only basic functionality.

2 Algorithmic results

We consider a given starting grid configuration C_s of unlabeled particles that needs to be transformed into a target configuration C_t by a sequence of simultaneous, collision-free motions in a minimum overall time, such that all intermediate configurations remain connected. The main algorithmic results illustrated in this video are as follows.

- It is NP-hard to decide whether C_s can be transformed into C_t within makespan 2.
- There is a constant c^* such that for any pair of start and target configurations with a (generalized) scale of at least c^* , a schedule with constant stretch can be computed in polynomial time.

The latter implies that there is a constant-factor approximation for the problem of computing schedules with minimal makespan restricted to pairs of start and target configurations with a scale of at least c^* .

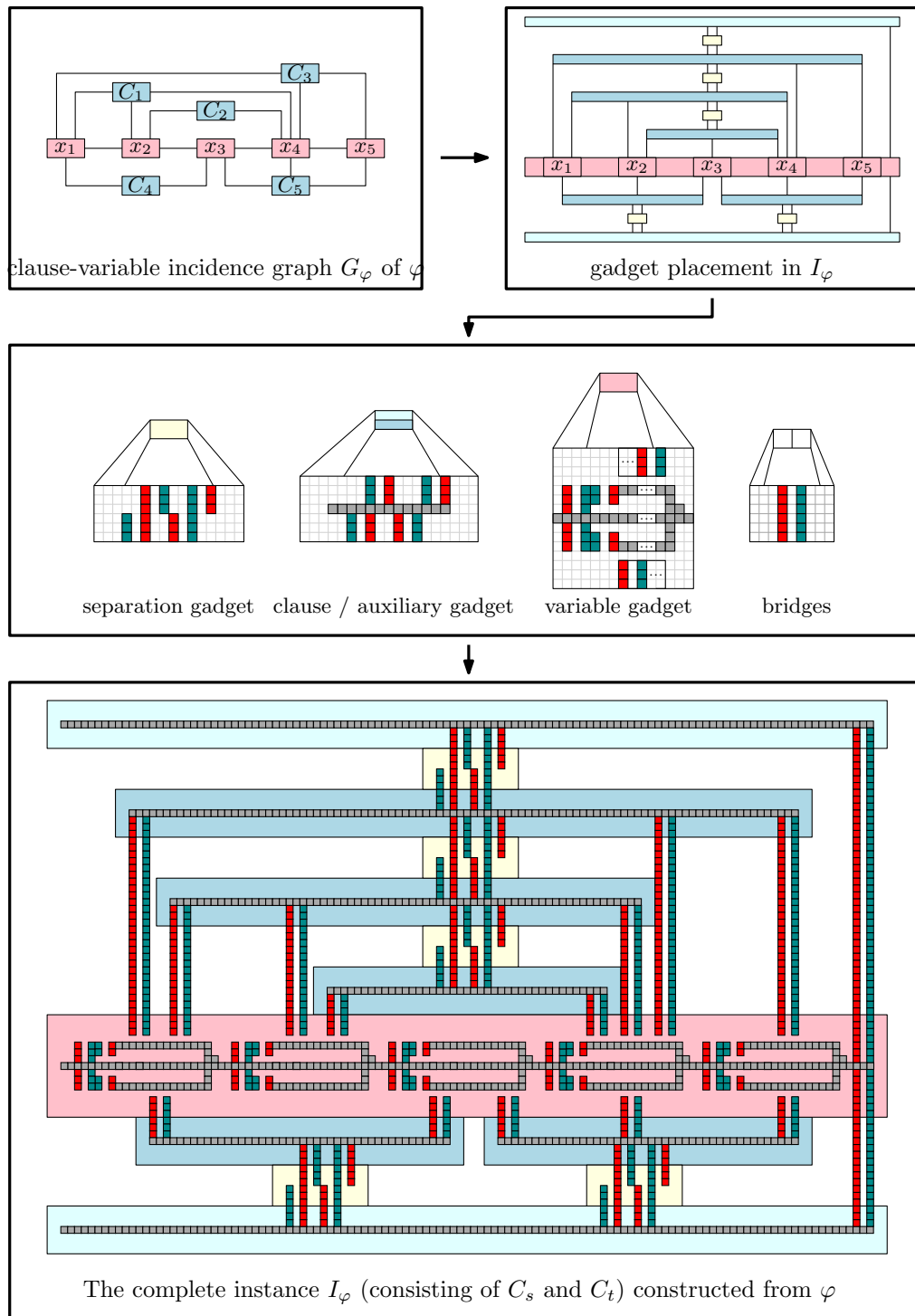
The hardness proof considers an instance φ of PLANAR MONOTONE 3SAT and constructs an instance I_φ with start configuration C_s and target configuration C_t ; see Figure 2, with start configuration (red), target configuration (dark cyan), and positions in both configurations (gray) indicated by colors. We consider a rectilinear planar embedding of the variable-clause incidence graph G_φ of φ , with variable vertices placed horizontally in a row, and clauses with unnegated and negated literals placed above and below, respectively. Variables of φ are represented by horizontal *variable gadgets* (light red). Two additional *auxiliary gadgets* (light blue) are positioned at the top and at the bottom boundary of the instance, connected to the variable gadget via bridges at the right boundary, and a *separation gadget* (yellow) between each adjacent and nested pair of *clause gadgets* (blue). All clause gadgets are connected via bridges to separation gadgets and possibly to the auxiliary gadgets. Further, there are bridges from a clause gadget to the respectively contained variables.

The overall approach for computing constant-stretch schedules works as follows; see Figure 3 (Top). In two preprocessing phases, we first ensure that the pair (C_s, C_t) overlaps in at least one position. For this, we move C_s towards C_t along a bottleneck matching such that the respective positions that realize the bottleneck distance, coincide. The overlap is necessary to successfully construct the auxiliary structure in the third phase of our approach. Afterwards, we use another bottleneck matching for mapping the start configuration C_s to the target configuration C_t , minimizing the maximum distance d between a start and a target location. Furthermore, we establish the scale in both configurations, set c to be the minimum of both scale values, and compute a suitable tiling whose tile size is $c \cdot d$, and that contain both C_s and C_t .

In a third phase, we build a scaffolding structure around C_s and C_t , based on the boundaries of cd -tiles of the specific tiling, see Figure 3 (Bottom). This provides connectivity throughout the actual reconfiguration. Restricting robot motion to their current and adjacent tiles also ensures constant stretch. Note that, as the size of the tiles is related to d , the scaffolding structure is connected.

In a fourth phase, we perform the actual reconfiguration of the arrangement. This consists of refilling the tiles of the scaffolding structure, achieving the proper number of robots within each tile, based on elementary flow computations. As a subroutine, we transform the robots inside each tile into a canonical “triangle” configuration, see Figure 3 (Top right).

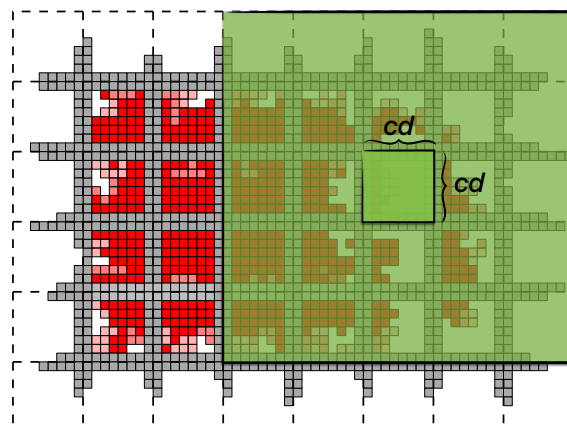
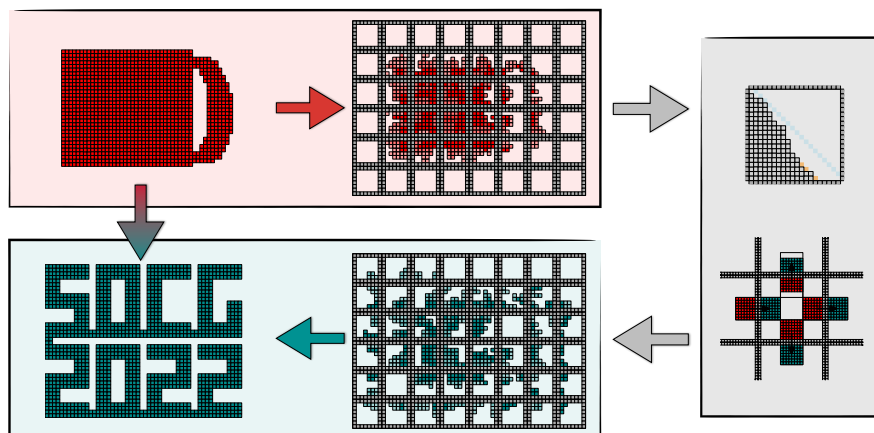
In a fifth and final phase, we disassemble the scaffolding structure and move the involved robots to their proper destinations.



■ **Figure 2** Symbolic overview of the NP-hardness reduction. The depicted instance is due to the PLANAR MONOTONE 3SAT formula $\varphi = (x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_4 \vee x_5) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_5)$. We use three different colors to indicate occupied positions in the start configuration (red), in the target configuration (dark cyan), and in both configurations (gray).

3 The video

The video starts with a description of the basic challenge, followed by real-world demonstrations of Catoms, their abilities to perform local reconfiguration and build large-scale structures, subject to maintaining connectivity. Then the idea and components of the hardness proof are shown. Finally, we provide a detailed animated description of the algorithmic method for achieving connected reconfiguration with bounded stretch for scaled arrangements, based on scaffold construction, flow computation and shifts between neighboring tiles, canonical triangle transformations within tiles, and scaffold removal.



■ **Figure 3** (Top) The algorithmic approach for achieving constant stretch while maintaining connectivity. (Bottom) Idea of the scaffold construction and tile size.

References

- 1 Amira Abdel-Rahman, Aaron T. Becker, Daniel Biediger, Kenneth C. Cheung, Sándor P. Fekete, Neil A. Gershenfeld, Sabrina Hugo, Benjamin Jenett, Phillip Keldenich, Eike Niehs, Christian Rieck, Arne Schmidt, Christian Scheffer, and Michael Yannuzzi. Space Ants: Constructing and reconfiguring large-scale structures with finite automata. In *Symposium on Computational Geometry (SoCG)*, pages 73:1–73:6, 2020. Video at <https://youtu.be/SFI5715d0vk>. doi: 10.4230/LIPIcs.SoCG.2020.73.
- 2 Aaron T. Becker, Sándor P. Fekete, Phillip Keldenich, Matthias Konitzny, Lillian Lin, and Christian Scheffer. Coordinated motion planning: The video. In *Symposium on Computational Geometry (SoCG)*, pages 74:1–74:6, 2018. Video at <https://www.ibr.cs.tu-bs.de/users/fekete/Videos/CoordinatedMotionPlanning.mp4>. doi:10.4230/LIPIcs.SoCG.2018.74.
- 3 Loïc Crombez, Guilherme Dias da Fonseca, Yan Gerard, Aldo Gonzalez-Lorenzo, Pascal Lafourcade, and Luc Libralesso. Shadoks approach to low-makespan coordinated motion planning. In *Symposium on Computational Geometry (SoCG)*, pages 63:1–63:9, 2021. doi: 10.4230/LIPIcs.SoCG.2021.63.
- 4 Erik D. Demaine, Sándor P. Fekete, Phillip Keldenich, Christian Scheffer, and Henk Meijer. Coordinated motion planning: Reconfiguring a swarm of labeled robots with bounded stretch. *SIAM Journal on Computing*, 48(6):1727–1762, 2019. doi:10.1137/18M1194341.
- 5 Sándor P. Fekete, Phillip Keldenich, Ramin Kosfeld, Christian Rieck, and Christian Scheffer. Connected coordinated motion planning with bounded stretch. In *Symposium on Algorithms and Computation (ISAAC)*, pages 9:1–9:16, 2021. doi:10.4230/LIPIcs.ISAAC.2021.9.
- 6 Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S. B. Mitchell. Computing coordinated motion plans for robot swarms: The CG:SHOP Challenge 2021, 2021. arXiv: 2103.15381.
- 7 Seth Copen Goldstein, Jason D. Campbell, and Todd C. Mowry. Programmable matter. *Computer*, 38(6):99–101, 2005. doi:10.1109/MC.2005.198.
- 8 Paul Liu, Jack Spalding-Jamieson, Brandon Zhang, and Da Wei Zheng. Coordinated motion planning through randomized k-opt. In *Symposium on Computational Geometry (SoCG)*, pages 64:1–64:8, 2021. doi:10.4230/LIPIcs.SoCG.2021.64.
- 9 Benoît Piranda and Julien Bourgeois. Designing a quasi-spherical module for a huge modular robot to create programmable matter. *Autonomous Robots*, 42(8):1619–1633, 2018. doi: 10.1007/s10514-018-9710-0.
- 10 Benoît Piranda and Julien Bourgeois. Datom: A deformable modular robot for building self-reconfigurable programmable matter. In *Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 70–81, 2021. doi:10.1007/978-3-030-92790-5_6.
- 11 David Soloveichik and Erik Winfree. Complexity of self-assembled shapes. *SIAM Journal on Computing*, 36(6):1544–1569, 2007. doi:10.1137/S0097539704446712.
- 12 Pierre Thalamy, Benoît Piranda, and Julien Bourgeois. Engineering efficient and massively parallel 3d self-reconfiguration using sandboxing, scaffolding and coating. *Robotics and Autonomous Systems*, 146:103875, 2021. doi:10.1016/j.robot.2021.103875.
- 13 Hyeyun Yang and Antoine Vigneron. A simulated annealing approach to coordinated motion planning. In *Symposium on Computational Geometry (SoCG)*, pages 65:1–65:9, 2021. doi: 10.4230/LIPIcs.SoCG.2021.65.

A Cautionary Tale: Burning the Medial Axis Is Unstable

Erin Chambers  

Saint Louis University, MO, USA

Christopher Fillmore  

IST Austria, Klosterneuburg, Austria

Elizabeth Stephenson  

IST Austria, Klosterneuburg, Austria

Mathijs Wintraecken  

IST Austria, Klosterneuburg, Austria

Abstract

The medial axis of a set consists of the points in the ambient space without a unique closest point on the original set. Since its introduction, the medial axis has been used extensively in many applications as a method of computing a topologically equivalent skeleton. Unfortunately, one limiting factor in the use of the medial axis of a smooth manifold is that it is not necessarily topologically stable under small perturbations of the manifold. To counter these instabilities various prunings of the medial axis have been proposed. Here, we examine one type of pruning, called burning. Because of the good experimental results, it was hoped that the burning method of simplifying the medial axis would be stable. In this work we show a simple example that dashes such hopes based on Bing’s house with two rooms, demonstrating an isotopy of a shape where the medial axis goes from collapsible to non-collapsible.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Medial axis, Collapse, Pruning, Burning, Stability

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.66

Category Media Exposition

Funding Partially supported by the DFG Collaborative Research Center TRR 109, “Discretization in Geometry and Dynamics” and the European Research Council (ERC), grant no. 788183, “Alpha Shape Theory Extended”.

Erin Chambers: Supported in part by the National Science Foundation through grants DBI-1759807, CCF-1907612, and CCF-2106672.

Mathijs Wintraecken: Supported by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 754411. The Austrian science fund (FWF) M-3073

Acknowledgements We thank André Lieutier, David Letscher, Ellen Gasparovic, Kathryn Leonard, and Tao Ju for early discussions on this work. We also thank Lu Liu, Yajie Yan and Tao Ju for sharing code to generate the examples.

1 Introduction

The medial axis $\text{ax}(\mathcal{S})$ of a closed set $\mathcal{S} \subset \mathbb{R}^d$ is the set of points in \mathbb{R}^d for which the closest point in \mathcal{S} is not unique. We note that although Federer [22] already studied the (complement of the) medial axis, the name was coined later by Blum [8]. The medial axis is used in many applications as a method of computing a topologically equivalent skeleton. The medial axis also has deep connections to singularity theory [2, 9, 17, 28, 29, 34, 35, 38].



© Erin Chambers, Christopher Fillmore, Elizabeth Stephenson, and Mathijs Wintraecken; licensed under Creative Commons License CC-BY 4.0

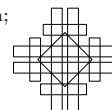
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 66; pp. 66:1–66:9

Leibniz International Proceedings in Informatics

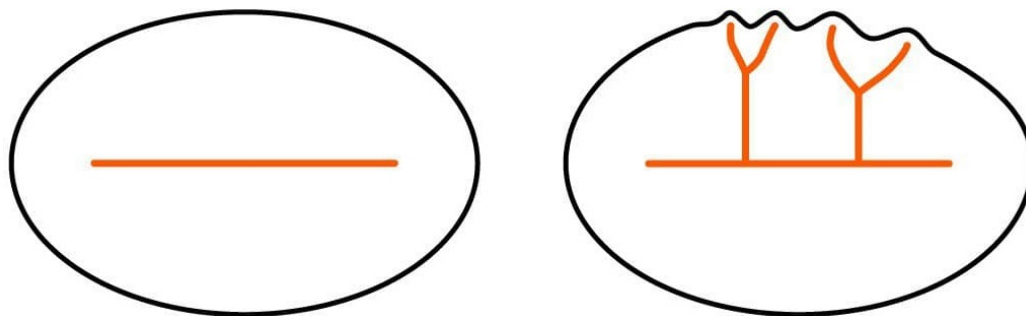


LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

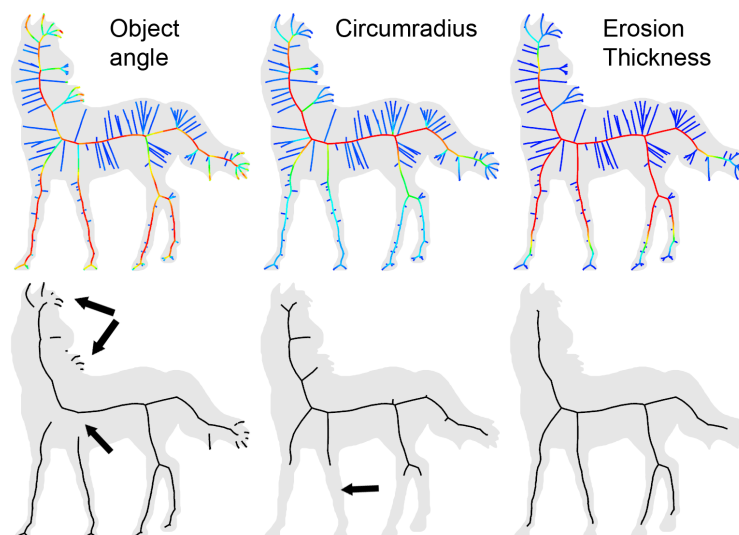


66:2 A Cautionary Tale: Burning the Medial Axis Is Unstable

Unfortunately, one limiting factor in the use of the medial axis is its (topological) instability under small perturbations [3]. Here small is understood to be small with respect to the Hausdorff distance. See Figure 1 for a standard example of such an instability. However the medial axis does capture the homotopy type [26, 36]. The radius function and medial axis together suffice to reconstruct the original set (under reasonable assumptions) [14, 15, 16, 18].



■ **Figure 1** Small perturbations (with respect to the Hausdorff distance) can lead to large perturbations of the medial axis.



■ **Figure 2** Various pruning methods, from left to right: Object angles [4, 21], radius of the set of closest points [12] (the λ -medial axis, also used in our computation), and a burning method proposed in [37], with various undesirable features indicated. The value of the object angle, radius of the set of closest points, and burning time is indicated in colour on top. Reproduced from [37].

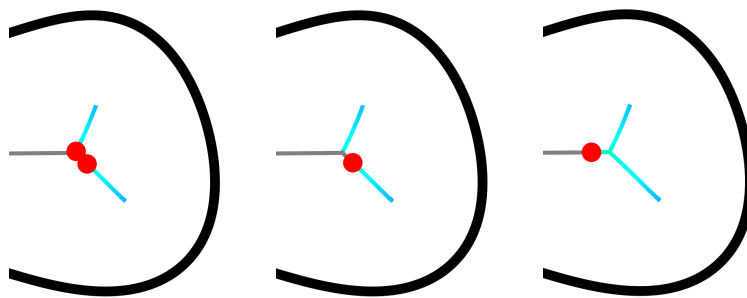
If we restrict ourselves to a smaller class of spaces and perturbations, stability results are available: Chazal and Soufflet [11] proved that the medial axis is stable with respect to the Hausdorff distance under ambient diffeomorphisms, if we assume that the set of positive reach is a C^2 manifold and the distortion is a C^2 diffeomorphism of \mathbb{R}^b .

Significant effort has gone into the simplification (pruning) of the medial axis. This was motivated by applications in graphics (where it is used as a skeleton, see the surveys [30, 33]), data reduction, shape recognition, and learning (see for example [5, 10, 19, 25, 27, 31, 32, 37]).

Many prunings of the medial axis have been proposed in many different settings [1, 4, 7, 12, 20, 21, 24, 27, 32]. See Figure 2 for an illustration of some commonly used ones and their pitfalls: the object angle, which is of historic importance in the community but can disconnect the medial axis, the λ -medial axis, which is used to compute a close approximation of the medial axis but which can truncate “thin” regions undesirably, and the burning method which we consider in this work.

2 Burning Bing’s house

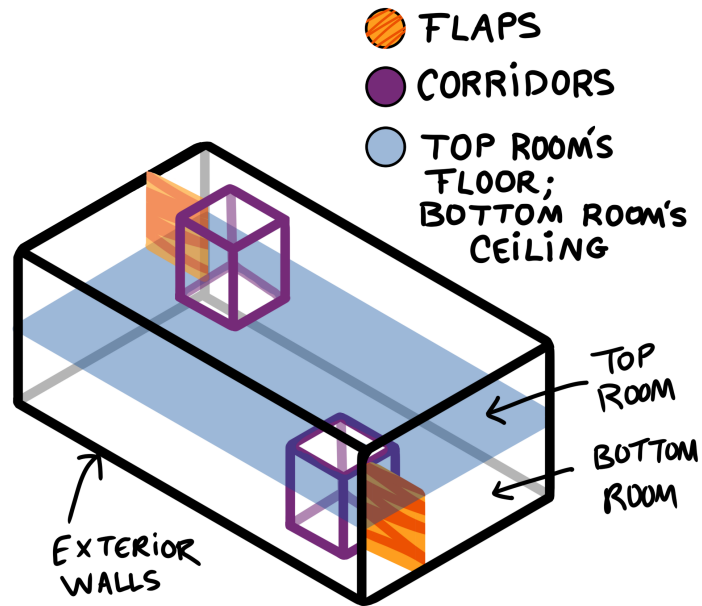
The simplification which we focus on for this work is the burning of the medial axis [37], which generalizes Blum’s original “grassfire” analogy for the medial axis. The burning of the medial axis removes the extremities of the medial axis by “starting a fire” at the boundary of the medial axis which stops if the fire hits an obstacle, as illustrated in Figure 3.



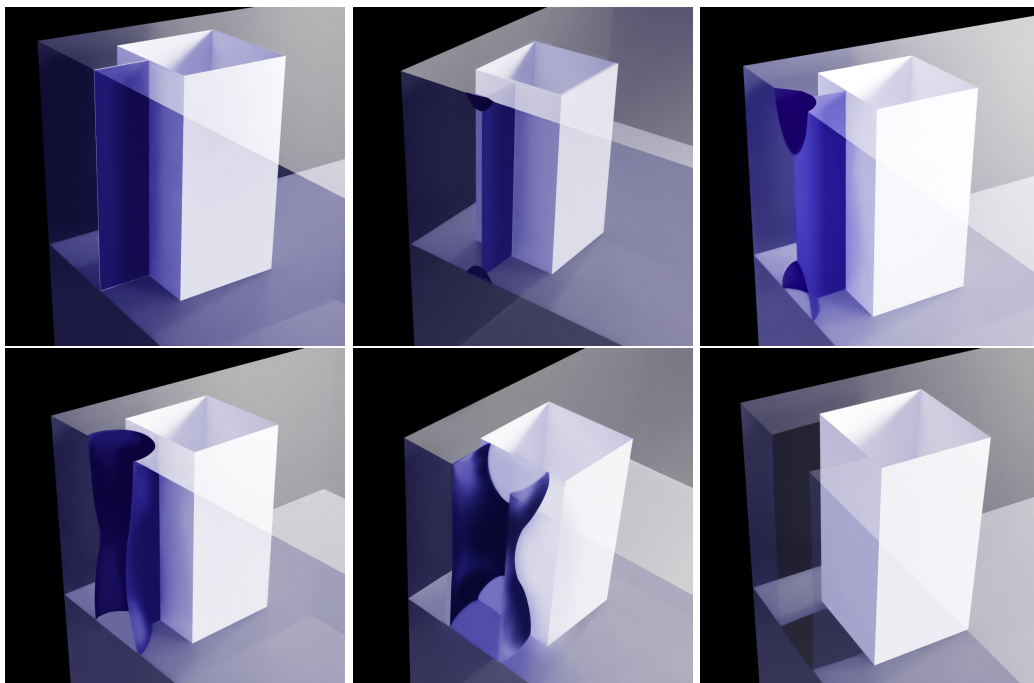
■ **Figure 3** The fire front progression on a the medial axis (grey) of a curve (black). As the fire front (indicated by the red dot) hits an unburned junction, it stops. If the junction is already burned (with the colour indicating the burn time) the fire continues.

Because of the good experimental results, it was conjectured that the burning method of simplification of the medial axis would be stable [27], i.e. no discontinuous jumps. In this work we show that this is not the case. The counter example is based on the standard deformation retract from the closed ball to Bing’s house with two rooms [6], which is a contractible but not collapsible two dimensional simplicial complex, see Figure 6. Bing’s house is not collapsible, as there is no boundary.

Before we go into the main statement we consider a deformation of Bing’s house which makes it collapsible. This deformation will be mirrored in the medial axis in our construction, this deformation is depicted in Figure 5, see Figure 4 for the nomenclature. In this construction we cut a flap open so that the room no longer completely runs around the corridor. This cutting exposes an edge of one of the walls of the corridor and path that goes from the edge to the bottom room. We can use this edge to collapse along the path into the bottom room, then the room, and from this the rest of Bing’s house.



■ Figure 4 The various parts of Bing's house indicated.



■ Figure 5 This deformation, which cuts a flap open, makes Bing's house collapsible.

The precise result is the following:

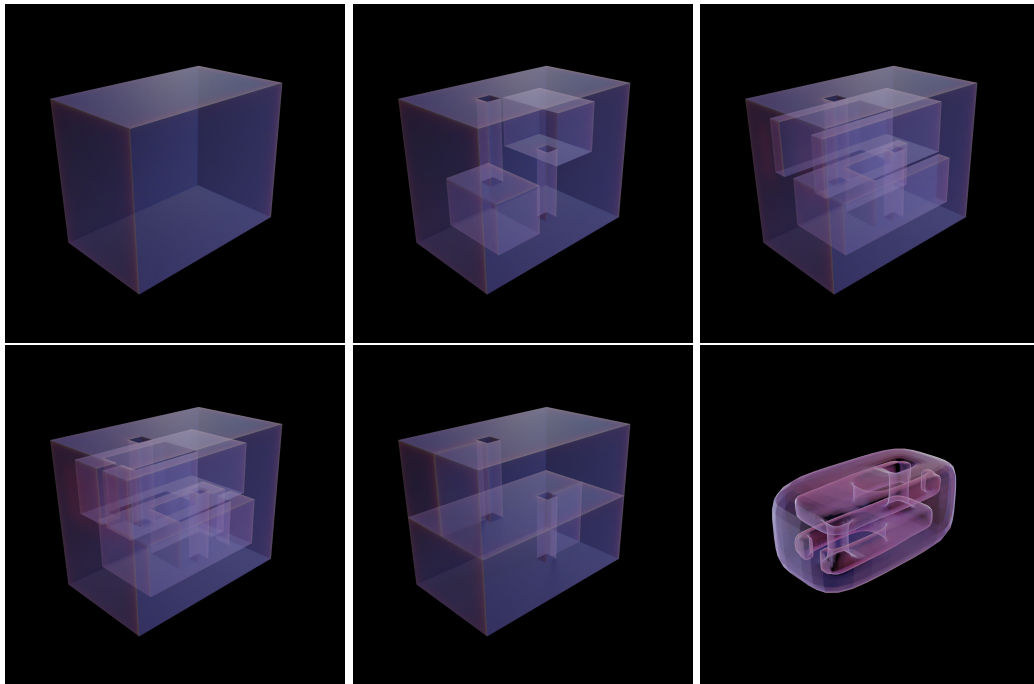
- **Theorem 1.** *There exists a smooth ambient isotopy $H_t : [0, 1] \times \mathbb{S}^2 \rightarrow \mathbb{R}^3$ such that:*
- *The medial axis $\text{ax}(H_0(\mathbb{S}^2))$ is collapsible/burn to a single point.*
 - *The medial axis $\text{ax}(H_1(\mathbb{S}^2))$ is Bing’s house and is therefore non-collapsible/ cannot burn.*
 - *The burning of $\text{ax}(H_t(\mathbb{S}^2))$ is not continuous in t with respect to the Hausdorff distance.*
 - *The topology of the burned axis changes from a point to Bing’s house with two rooms at a single $t_0 \in [0, 1]$.*
 - *The isotopy H_t can be chosen to be generic in the sense of singularity theory as developed by Arnol’d and Thom [2], see in particular [23].*

Proof. Bing constructed his house as a deformation retract from a solid cube, see Figure 6. The isotopy of the sphere we consider is the boundary of this deformation. However instead of reducing to a two dimensional object we skip the last step so that every point in the deformation the set remains a topological (solid) ball and its boundary a sphere. The end point of this deformation is a thickened version Bing’s house. We will only consider the medial axis in the interior of the sphere and not the exterior. The medial axis of a thickened version of Bing’s house is Bing’s house itself. The deformation is depicted in Figure 7. The essential topological change only happens near the end of the deformation when the room wraps around the corridor, see Figure 8. When the bisector between the corridor and the wall disappears and is replaced by the bisector between the two parts of the room that are wrapping around the corridor, the medial axis becomes non-collapsible. This transition can be made generic in terms of the transitions of the singularities [23]. ◀

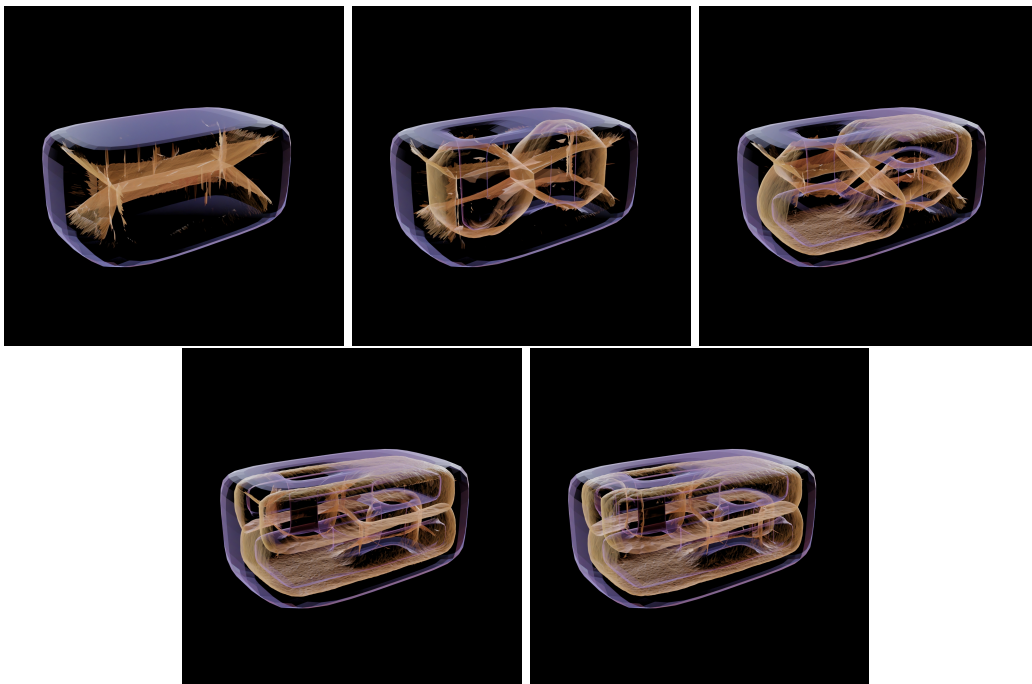
We illustrate this deformation in our video; see also Figures 6, 7, and 8. These animations were made using the λ -medial axis (see https://github.com/cdfillmore/lambda_medial_axis) and the open source software Blender [13]. Here λ is chosen very small to ensure that the λ -medial axis is a good approximation of the medial axis.

- **Corollary 2.** *Collapsing or pruning the medial axis of a domain such that it becomes one-dimensional, as proposed in e.g. [7], is not always possible, even if the boundary of a domain is a smooth sphere.*

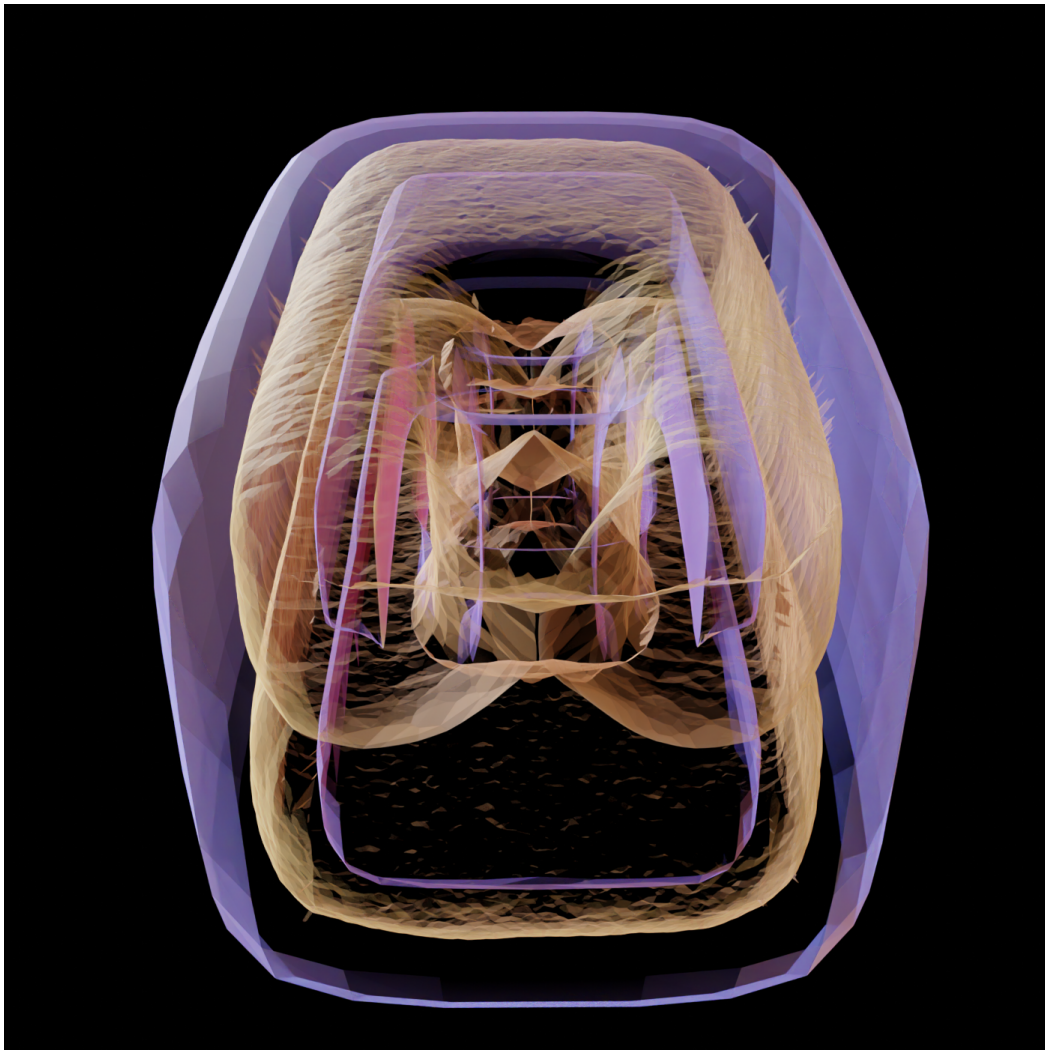
66:6 A Cautionary Tale: Burning the Medial Axis Is Unstable



■ **Figure 6** The deformation retract of a solid cube (topological ball) to Bing's house. In the final frame we show the smoothed version of a thickened Bing's house used in the computation.



■ **Figure 7** The evolution of the medial axis (yellow) in the interior as the solid cube is deformed into a thickened version of Bing's house (blue/purple).



■ **Figure 8** The critical transition of the medial axis. There are points on the medial axis equidistant to the two parts of the room that wrap around the corridor, the corridor itself and the exterior wall, which can be avoided by a small perturbation. This transition occurs between frames 4 and 5 of Figure 6.

References

- 1 Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2-3):127–153, 2001. doi:10.1016/S0925-7721(01)00017-7.
- 2 Vladimir Arnol'd. *Singularities of caustics and wave fronts*, volume 62 of *Mathematics and its Applications*. Springer Science & Business Media, 2013.
- 3 D. Attali, J.-D. Boissonnat, and H. Edelsbrunner. Stability and computation of medial axes - a state-of-the-art report. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Mathematics and Visualization, pages 109–125. Springer Berlin Heidelberg, 2009.
- 4 Dominique Attali and Annick Montanvert. Modeling noise for a better simplification of skeletons. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 13–16. IEEE, 1996. doi:10.1109/ICIP.1996.560357.

- 5 Gulce Bal, Julia Diebold, Erin Wolf Chambers, Ellen Gasparovic, Ruizhen Hu, Kathryn Leonard, Matineh Shaker, and Carola Wenk. Skeleton-based recognition of shapes in images via longest path matching. In *Research in Shape Modeling*, volume 1 of *Association for Women in Mathematics Series*, pages 81–99. Springer, 2015. doi:10.1007/978-3-319-16348-2_6.
- 6 RH Bing. Some aspects of the topology of 3-manifolds related to the Poincaré conjecture. In T.L. Saaty, editor, *Lectures on modern mathematics*, volume II, pages 93–128. John Wiley and Sons, 1964.
- 7 Thibault Blanc-Beyne, Géraldine Morin, Kathryn Leonard, Stefanie Hahmann, and Axel Carlier. A salience measure for 3D shape decomposition and sub-parts classification. *Graphical Models*, 99:22–30, 2018. doi:10.1016/j.gmod.2018.07.003.
- 8 Harry Blum. A Transformation for Extracting New Descriptors of Shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.
- 9 Michael A. Buchner. The structure of the cut locus in dimension less than or equal to six. *Compositio Mathematica*, 37(1):103–119, 1978. URL: http://www.numdam.org/item/CM_1978__37_1_103_0/.
- 10 Ming-Ching Chang and Benjamin B. Kimia. Measuring 3d shape similarity by graph-based matching of the medial scaffolds. *Computer Vision and Image Understanding*, 115(5):707–720, May 2011. doi:10.1016/j.cviu.2010.10.013.
- 11 F. Chazal and R. Soufflet. Stability and finiteness properties of medial axis and skeleton. *Journal of Dynamical and Control Systems*, 10(2):149–170, 2004. doi:10.1023/B:JODS.0000024119.38784.ff.
- 12 Frédéric Chazal and André Lieutier. The “ λ -medial axis”. *Graphical Models*, 67(4):304–331, 2005. doi:10.1016/j.gmod.2005.01.002.
- 13 Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL: <http://www.blender.org>.
- 14 James Damon. Smoothness and geometry of boundaries associated to skeletal structures I: Sufficient conditions for smoothness. In *Annales de l’institut Fourier*, volume 53(6), pages 1941–1985, 2003. doi:10.5802/aif.1997.
- 15 James Damon. Smoothness and geometry of boundaries associated to skeletal structures, II: Geometry in the Blum case. *Compositio Mathematica*, 140(6):1657–1674, 2004. doi:10.1112/S0010437X04000570.
- 16 James Damon. Determining the geometry of boundaries of objects from medial data. *International Journal of Computer Vision*, 63(1):45–64, 2005.
- 17 James Damon. The global medial structure of regions in \mathbb{R}^3 . *Geometry & Topology*, 10(4):2385–2429, 2006. doi:10.2140/gt.2006.10.2385.
- 18 James Damon. Global geometry of regions and boundaries via skeletal and medial integrals. *Communications in Analysis and Geometry*, 15(2):307–358, 2007. doi:10.4310/CAG.2007.v15.n2.a5.
- 19 Ilke Demir, Camilla Hahn, Kathryn Leonard, Geraldine Morin, Dana Rahbani, Athina Panotopoulou, Amelie Fondevilla, Elena Balashova, Bastien Durix, and Adam Kortylewski. SkelNetOn 2019: Dataset and challenge on deep learning for geometric shape understanding. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1143–1151, 2019. doi:10.1109/CVPRW.2019.00149.
- 20 Tamal K. Dey and Jian Sun. Defining and computing curve-skeletons with medial geodesic function. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP ’06*, pages 143–152, Goslar, DEU, 2006. Eurographics Association.
- 21 Tamal K Dey and Wulue Zhao. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica*, 38(1):179–200, 2004. doi:10.1007/s00453-003-1049-y.
- 22 H. Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93:418–491, 1959. doi:10.1090/S0002-9947-1959-0110078-1.

- 23 Peter J Giblin, Benjamin B Kimia, and Anthony J Pollitt. Transitions of the 3D medial axis under a one-parameter family of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):900–918, 2008. doi:10.1109/TPAMI.2008.120.
- 24 Joachim Giesen, Balint Miklos, Mark Pauly, and Camille Wormser. The scale axis transform. In *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, pages 106–115, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1542362.1542388.
- 25 Seng-Beng Ho and Charles R Dyer. Shape smoothing using medial axis properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):512–520, 1986. doi:10.1109/TPAMI.1986.4767815.
- 26 André Lieutier. Any open bounded subset of \mathbb{R}^n has the same homotopy type as its medial axis. *Computer-Aided Design*, 36(11):1029–1046, 2004. Solid Modeling Theory and Applications. doi:10.1016/j.cad.2004.01.011.
- 27 Lu Liu, Erin W. Chambers, David Letscher, and Tao Ju. Extended grassfire transform on medial axes of 2d shapes. *Computer-Aided Design*, 43(11):1496–1505, 2011. Solid and Physical Modeling 2011. doi:10.1016/j.cad.2011.09.002.
- 28 Eduard Looijenga. *Structural Stability of smooth families of C^∞ -functions*. PhD thesis, Universiteit van Amsterdam, 1974.
- 29 John N Mather. Distance from a submanifold in Euclidean-space. In *Proceedings of symposia in pure mathematics*, volume 40, pages 199–216. American Mathematical Society, 1983.
- 30 Punam K Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern recognition letters*, 76:3–12, 2016.
- 31 T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571, 2004. doi:10.1109/TPAMI.2004.1273924.
- 32 Doron Shaked and Alfred M. Bruckstein. Pruning medial axes. *Computer Vision and Image Understanding*, 69(2):156–169, 1998. doi:10.1006/cviu.1997.0598.
- 33 Andrea Tagliasacchi, Thomas Delame, Michela Spagnuolo, Nina Amenta, and Alexandru Telea. 3D skeletons: A state-of-the-art report. In *Computer Graphics Forum*, volume 35(2), pages 573–597. Wiley Online Library, 2016. doi:10.1111/cgf.12865.
- 34 R. Thom. Sur le cut-locus d’une variété plongée. *Journal of Differential Geometry*, 6(4):577–586, 1972. doi:10.4310/jdg/1214430644.
- 35 Martijn van Manen. Maxwell strata and caustics. In *Singularities In Geometry And Topology*, pages 787–824. World Scientific, 2007.
- 36 Franz-Erich Wolter. Cut locus and medial axis in global shape interrogation and representation. In *MIT Design Laboratory Memorandum 92-2 and MIT Sea Grant Report*, 1992.
- 37 Yajie Yan, Kyle Sykes, Erin Chambers, David Letscher, and Tao Ju. Erosion thickness on medial axes of 3d shapes. *ACM Transactions on Graphics*, 35(4):38:1–38:12, July 2016. doi:10.1145/2897824.2925938.
- 38 Yosef Yomdin. On the local structure of a generic central set. *Compositio Mathematica*, 43(2):225–238, 1981. URL: http://www.numdam.org/item/CM_1981__43_2_225_0/.


Visualizing and Unfolding Nets of 4-Polytopes

Satyan L. Devadoss 

Department of Mathematics, University of San Diego, CA, USA

Matthew S. Harvey 

Department of Mathematics and Computer Science,
University of Virginia's College at Wise, VA, USA

Sam Zhang  

Department of Applied Mathematics, University of Colorado Boulder, CO, USA

Abstract

Over a decade ago, it was shown that every edge unfolding of the Platonic solids was without self-overlap, yielding a valid net. Recent work has extended this property to their higher-dimensional analogs: the 4-cube, 4-simplex, and 4-orthoplex. We present an interactive visualization that allows the user to unfold these polytopes by drawing on their dual 1-skeleton graph.

2012 ACM Subject Classification Theory of computation → Computational geometry; Applied computing → Computer-assisted instruction

Keywords and phrases unfoldings, nets, polytopes

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.67

Category Media Exposition

Supplementary Material *Software (Web-Application)*: <https://sam.zhang.fyi/html/unfolding/>

Funding *Sam Zhang*: Supported by an NSF Graduate Research Fellowship Award DGE 2040434.

1 Unfolding Polytopes

The study of unfolding polyhedra was popularized by Albrecht Dürer in the early 16th century who first recorded examples of polyhedral *nets*, connected edge unfoldings of polyhedra that lay flat on the plane without overlap. Motivated by this, Shephard [8] conjectured that every convex polyhedron can be cut along certain edges to admit a net. This claim remains open.

Consider this question for higher-dimensional *polytopes*: The codimension-one faces of a polytope are *facets* and its codimension-two faces are *ridges*. The analog of an edge unfolding of polyhedron is the *ridge unfolding* of an n -dimensional polytope: the process of cutting the polytope along a collection of its ridges so that the resulting (connected) arrangement of its facets develops isometrically into an \mathbb{R}^{n-1} hyperplane. Such an unfolding without overlap of its facets yields a valid *net*. Instead of trying to find one net for each convex polyhedron (as posed by Shephard), we consider a more aggressive property:

► **Definition 1.** *A polytope P is all-net if every ridge unfolding of P yields a valid net.*

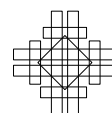
A decade ago, Horiyama and Shoji [7] showed that the five Platonic solids are all-net. Recent work [4] has shown applications in protein science: polyhedral nets are used to find a balance between entropy loss and energy gain for the folding propensity of a given shape. The higher-dimensional analogs of the Platonic solids are the regular polytopes. Three classes of regular polytopes exist for all dimensions: the n -simplex, n -cube, and n -orthoplex (sometimes called the *cross-polytope* or the *cocube*). The following is from [2] and [3]:

► **Theorem 2.** *The 4-simplex, the 4-cube, and the 4-orthoplex are all-net.*



© Satyan L. Devadoss, Matthew S. Harvey, and Sam Zhang;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 67; pp. 67:1–67:4
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

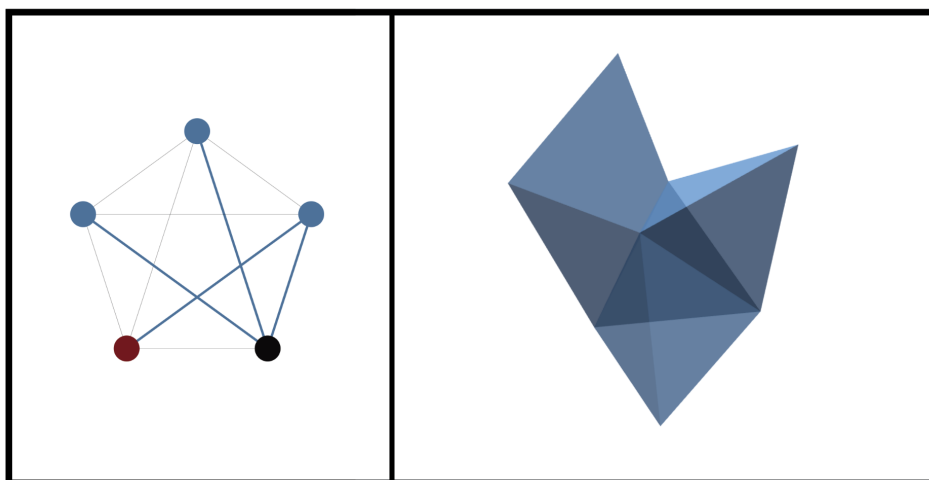


► **Remark 3.** For $n > 4$, the n -simplex and n -cube are all-net, while the n -orthoplex fails. Three additional regular polytopes appear only in four-dimensions: the 24-cell, 120-cell, and 600-cell. Their all-net property remain unexplored.

A ridge unfolding of a convex 4-dimensional polytope is given by a series of cuts along its 2-dimensional ridges so that the polytope may be unwrapped and “laid flat” in \mathbb{R}^3 . The goal of our visualization is to show the resulting net – the final placement of the unwrapped facets – rather than the unwrapping itself. Such an unfolding is specified by the combinatorics of the arrangement of its facets in the resulting net. In particular, a ridge unfolding of polytope P induces a *spanning tree* in the 1-skeleton of the dual of P : a tree whose nodes are the facets of the polytope and whose edges are the uncut ridges between the facets.

We now consider these associated graphs, the 1-skeleton of the duals of these polytopes: Since the 4-simplex is self-dual, its 1-skeleton is simply the *complete graph on 5 nodes* (corresponding to the 5 facets of the 4-simplex). The 4-cube is dual to the 4-orthoplex, whose 1-skeleton forms the *4-Roberts graph*. The 8 nodes of this graph can arranged on a circle so that antipodal nodes represent opposite facets of the cube. Finally, the dual of the 4-orthoplex is the 4-cube, whose 1-skeleton forms the *4-hypercube graph*. We chose a drawing of this graph where its 16 nodes are arranged on a circle.

The work of Buekenhout and Parker [1] has enumerated the spanning trees on these three graphs. Since unfoldings are in bijection with spanning trees, there are (up to symmetry), 3 distinct unfoldings of the 4-simplex, 261 distinct unfoldings of the 4-cube, and 110,912 distinct unfoldings of the 4-orthoplex. By Theorem 2 above, each of these unfoldings is a valid net. Our visualization software (<https://sam.zhang.fyi/html/unfolding/>) allows the user to interactively create all of these nets. The figures in this paper show three examples.

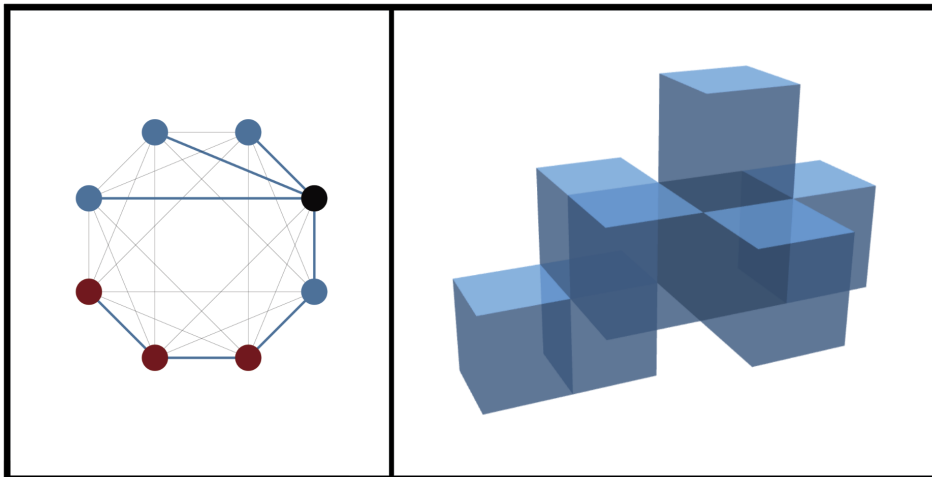


■ **Figure 1** A user-drawn spanning tree and its corresponding unfolded 4-simplex net.

2 Unfolding Geometry

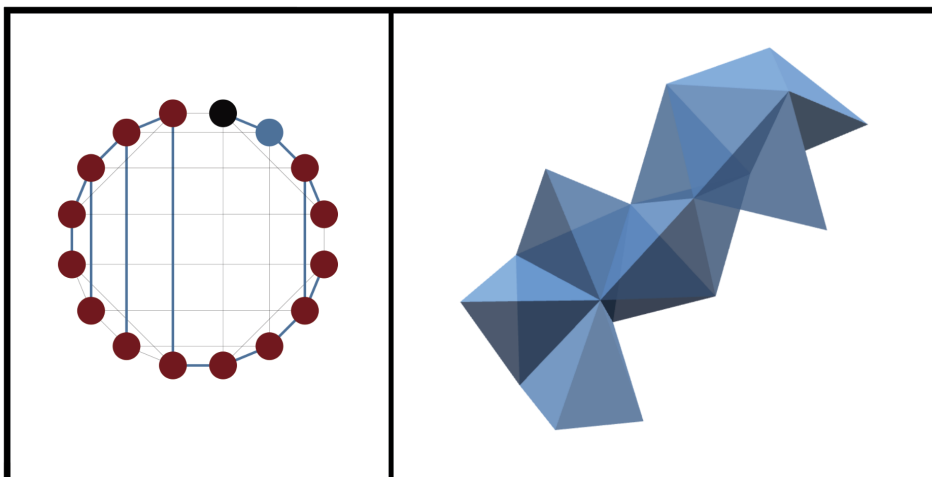
An unfolding is specified, step-by-step, by drawing a spanning tree. As it is being drawn, the corresponding net is formed by attaching new facets along the faces indicated by the tree.

In the case of the hypercube, the facets are cubes. The first cube is placed with its center (centroid) at the origin and its faces parallel to the coordinate planes. Each subsequent facet is attached to an exposed face f of one of the facets F in the existing structure as follows: the center P of F is translated one edge length in the direction perpendicular to F , to a new point Q (so that f bisects PQ). A new facet is then placed with Q as its center.



■ **Figure 2** A user-drawn spanning tree and its corresponding unfolded 4-cube net.

In the case of the simplex and the orthoplex, the facets are tetrahedra. Unlike the cube, a tetrahedron cannot be conveniently embedded in \mathbb{R}^3 , making calculations there difficult. It can be much more elegantly placed in \mathbb{R}^4 , with its vertices at $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$, and $(0, 0, 0, 1)$. Each subsequent facet is then attached to an exposed face f of a facet F by reflection across f . This reflection will fix all of f , hence all the vertices of F except for one, say P . Thanks to the \mathbb{R}^4 embedding, its reflection, Q , can be calculated by a simple matrix multiplication. A new facet is then constructed whose vertices are those of f , along with Q . In this construction, the unfolded net will lie in the hyperplane $x_1 + x_2 + x_3 + x_4 = 1$. Once vertex coordinates have been calculated, it is necessary to rotate the shape into standard 3-dimensional space $x_4 = 0$ before displaying the final result.



■ **Figure 3** A user-drawn spanning tree and its corresponding unfolded 4-orthoplex net.

3 Implementation

Our visualization is an interactive, open source browser application implemented using HTML5 and JavaScript, and the application and source code can be accessed at <https://sam.zhang.fyi/html/unfolding/>. The user can select whether to unfold the 4-cube, the 4-simplex, or the 4-orthoplex. The user performs the unfolding by drawing a spanning tree on the graph of the 1-skeleton of the dual polytope. The graph of the 1-skeleton is represented using the JavaScript library JSXGraph [6], and the unfolded faces in \mathbb{R}^3 are drawn in WebGL using ThreeJS [5]. We used built-in features of ThreeJS to allow the user to scroll to zoom and click and drag to rotate the unfolded object.

The architecture of the application reuses common components for unfolding the cube, simplex, and orthoplex. In particular, we implement our own spanning tree data structure, which together with the underlying structure of the graph of the 1-skeleton of the dual polytope allows us to determine the set of valid moves. We maintain an undo stack of a single move, so that the visualization displays the outcome of a move when a valid node on the 1-skeleton is moused over, and saves the move if and only if a click is registered on the node before the mouse is moved off of the node. Otherwise, the move is undone when the mouse leaves the node.

There are a variety of choices for embedding the 1-skeleton of the dual polytope onto the plane, though we can pick elegant choices that position all of the nodes around the circumference of a circle. For the simplex, we have a standard visualization of a clique, and for the cube, we draw the 1-skeleton (a Roberts graph) as a clique with the opposite edges removed. For the orthoplex, we embed its 1-skeleton in a way that all of the edges either form part of the “circumference” of the graph or are parallel to the plane’s vertical and horizontal axes.

We emphasize the current node by highlighting it as black on both the 1-skeleton as well as in the unfolding. Unvisited nodes are colored blue (if accessible) and red (if inaccessible), while visited nodes are shaded dark blue and dark red, appropriately. We arbitrarily fix a node as the starting one. Due to the ability of the user to pan the camera around the unfolding, all unfoldings up to rotation, but not reflection, are identified in the visualization. We introduced a minor amount of transparency into the unfolding so that the user can more clearly see the structure of the overall object.

References

- 1 Francis Buekenhout and Monique Parker. The number of nets of the regular convex polytopes in dimension ≤ 4 . *Discrete mathematics*, 186:69–94, 1998.
- 2 Kristin DeSplinter, Satyan L Devadoss, Jordan Readyhough, and Bryce Wimberly. Unfolding cubes: Nets, packings, partitions, chords. *Electronic Journal of Combinatorics*, 27:4–41, 2020.
- 3 Satyan L Devadoss and Matthew Harvey. Unfoldings and nets of regular polytopes. *arXiv*, 2021. [arXiv:2111.01359](https://arxiv.org/abs/2111.01359).
- 4 Paul Dodd, Pablo Damasceno, and Sharon Glotzer. Universal folding pathways of polyhedron nets. *Proceedings of the National Academy of Science*, 115:6690–6696, 2018.
- 5 Mr. Doob. ThreeJS, 2021. URL: <https://github.com/mrdoob/three.js>.
- 6 Michael Gerhäuser, Bianca Valentin, and Alfred Wassermann. JSXGraph: Dynamic mathematics with JavaScript. *International Journal for Technology in Mathematics Education*, 17(4), 2010.
- 7 Takashi Horiyama and Wataru Shoji. Edge unfoldings of platonic solids never overlap. In *Proceedings of the 23rd Canadian Conference on Computational Geometry*, 2011.
- 8 Geoffrey C Shephard. Convex polytopes with convex nets. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 78. Cambridge University Press, 1975.

Visualizing WSPDs and Their Applications

Anirban Ghosh  

School of Computing, University of North Florida, Jacksonville, FL, USA

FNU Shariful  

School of Computing, University of North Florida, Jacksonville, FL, USA

David Wisnosky  

School of Computing, University of North Florida, Jacksonville, FL, USA

Abstract

Introduced by Callahan and Kosaraju back in 1995, the concept of well-separated pair decomposition (WSPD) has occupied a special significance in computational geometry when it comes to solving distance problems in d -space. We present an in-browser tool that can be used to visualize WSPDs and several of their applications in 2-space. Apart from research, it can also be used by instructors for introducing WSPDs in a classroom setting. The tool will be permanently maintained by the third author at <https://wisno33.github.io/VisualizingWSPDsAndTheirApplications/>.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases well-separated pair decomposition, nearest neighbor, geometric spanners, minimum spanning tree

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.68

Category Media Exposition

Supplementary Material *Software (Web-Application)*:

<https://wisno33.github.io/VisualizingWSPDsAndTheirApplications/>

Funding Research on This Paper Is Supported by the NSF Award CCF-1947887.

1 Introduction

Let P and Q be two finite pointsets in d -space and s be a positive real number. We say that P and Q are well-separated with respect to s , if there exist two congruent disjoint balls B_P and B_Q , such that B_P contains the bounding-box of P , B_Q contains the bounding-box of Q , and the distance between B_P and B_Q is at least s times the common radius of B_P and B_Q . The quantity s is referred to as the *separation ratio* of the decomposition. Using this idea of well-separability, one can define a well-separated decomposition of a pointset (WSPD) [4] in the following way. Let P be a set of n points in d -space and s be a positive real number. A well-separated pair decomposition for P with respect to s is a collection of pairs of non-empty subsets of P , $\{A_1, B_1\}, \{A_2, B_2\}, \dots, \{A_m, B_m\}$ for some integer m (referred to as the size of the WSPD) such that

- for each i with $1 \leq i \leq m$, A_i and B_i are well-separated with respect to s , and
- for any two distinct points $p, q \in P$, there is exactly one index i with $1 \leq i \leq m$, such that $p \in A_i, q \in B_i$, or $p \in B_i, q \in A_i$.

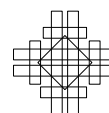
Note that in some cases, $m = C(n, 2) = \Theta(n^2)$. Refer to [5, 6, 7] for a detailed discussion on WSPDs and their uses. In this work, we consider WSPDs in 2-space only. Our implementations are based on the algorithms presented in the book by Narasimhan and Smid [6, Chapters 9 and 10]. These algorithms were originally presented in [2, 3, 4] by Callahan and Kosaraju.



© Anirban Ghosh, FNU Shariful, and David Wisnosky;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 68; pp. 68:1–68:4
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



2 Algorithms implemented

We have implemented the algorithms using the JSXGraph library. Some code segments have been borrowed from the tool presented in [1].

2.1 Constructing WSPDs

Given a pointset P and a positive real number s , a WSPD of P can be constructed using a split-tree. Our implementation is based on the naive quadratic time approach presented in [6]. It accepts P and s , and returns the WSPD pairs in the WSPD decomposition. Refer to Algorithm 1. An advanced linearithmic construction is also presented in [6].

Notations. Let x be a split-tree node. Let S_x denotes the points stored in the subtree rooted at x and $R(x)$ denotes the bounding-box of S_x . Further, $L_{max}(R(x))$ denotes the length of the longer side of $R(x)$.

■ **Algorithm 1** CONSTRUCTWSPD($P, s > 0$).

1. Construct a split-tree T on P in the following way:

If $|P| = 1$, then the split-tree consists of one single node that stores that point. Otherwise, split the bounding-box of P into two rectangles by cutting the longer side of the bounding-box into two equal parts. Let P_1 and P_2 be the two subsets of P that are contained in these two new rectangles. The split-tree for P consists of a root having two subtrees, which are recursively defined for P_1 and P_2 .

2. For each internal node u of T , find WSPD pairs using v and w , the left and right child of u , respectively, in the following way:
 - a. Compute $S_v, S_w, L_{max}(R(v))$ and $L_{max}(R(w))$.
 - b. If S_v, S_w are well-separated with respect to s , then node pair $\{v, w\}$ is a WSPD pair. Otherwise, if $L_{max}(R(v)) \leq L_{max}(R(w))$, recursively find WSPD pairs using v , LEFTCHILD(w) and then recursively find WSPD pairs using v , RIGHTCHILD(w). Else, recursively find WSPD pairs using LEFTCHILD(v), w , and then recursively find WSPD pairs using RIGHTCHILD(v), w .
-

2.2 Applications of WSPDs

- CONSTRUCTION OF t -SPANNERS. Given a pointset P and $t \geq 1$, a t -spanner on P is a Euclidean geometric graph G on P such that for every pair of points $p, q \in P$, the length of the shortest-path between p, q in G is at most t times the Euclidean distance between them. Refer to Algorithm 2. It returns the set of spanner edges and can be implemented to run in $O(n \log n)$ time [6].

■ **Algorithm 2** CONSTRUCT- t -SPANNER($P, t > 1$).

Let $s = 4(t + 1)/(t - 1)$. Construct a WSPD of P with separation ratio s . For every pair (A_i, B_i) of the decomposition do the following: include the edge $\{a_i, b_i\}$ in the spanner where a_i is an arbitrary point in A_i and b_i is an arbitrary point in B_i .

- FINDING CLOSEST PAIRS. The problem asks to find two distinct points of P whose distance is minimum among the $C(n, 2)$ point pairs. The idea of well-separatedness can be used to design an algorithm for this problem. See Algorithm 3. It can be implemented to run in $O(n \log n)$ time [6].

■ **Algorithm 3** CLOSESTPAIR(P).

Construct a 2-spanner using Algorithm 2. Since the closest pair is connected by an edge of the spanner, find the pair by iterating over all the edges.

- FINDING k -CLOSEST PAIRS. It is a generalization of the closest-pair problem. Given a positive integer k such that $k \leq C(n, 2)$, the goal is to find the k closest pairs among the $C(n, 2)$ pairs. See Algorithm 4. It can be implemented to run in $O(n \log n + k)$ time [6].

■ **Algorithm 4** k -CLOSESTPAIRS(P).

-
1. Create a WSPD with some $s > 0$. For every pair (A_i, B_i) in the decomposition, let $R(A_i)$ and $R(B_i)$ be the bounding boxes of A_i and B_i , respectively. Further, by $|R(A_i)R(B_i)|$, we denote the minimum distance between the two bounding-boxes $R(A_i), R(B_i)$. Renumber the m pairs in the decomposition such that $|R(A_1)R(B_1)| \leq |R(A_2)R(B_2)| \leq \dots \leq |R(A_m)R(B_m)|$.
 2. Compute the smallest integer $\ell \geq 1$, such that $\sum_{i=1}^{\ell} |A_i| \cdot |B_i| \geq k$.
 3. Let $r := |R(A_\ell)R(B_\ell)|$.
 4. Compute the integer ℓ' , which is defined as the number of indices with $1 \leq i \leq m$, such that $|R(A_i)R(B_i)| \leq (1 + 4/s)r$.
 5. Compute the set L consisting of all pairs $\{p, q\}$ for which there is an index i with $1 \leq i \leq \ell'$, such that $p \in A_i, q \in B_i$ or $q \in A_i, p \in B_i$.
 6. Compute and return the k smallest distances determined by the pairs in the set L .
-

- FINDING ALL-NEAREST NEIGHBORS. In this problem, for every point p in P , we need to find its nearest neighbor q in $P \setminus \{p\}$. Refer to Algorithm 5 for a description of the algorithm. It can be implemented to run in $O(n \log n)$ time [6].

■ **Algorithm 5** ALLNEARESTNEIGHBORS(P).

Choose $s > 2$ and obtain the pairs of WSPD. For every $p \in P$, compute its nearest neighbor in the following way: Find all such pairs of the WSPD, for which at least one of their sets is a singleton containing p . For every such pair (A_i, B_i) , if $A_i = \{p\}$, then $S_p = S_p \cup B_i$, else if $B_i = \{p\}$, then $S_p = S_p \cup A_i$. The nearest neighbor of p is the point in S_p closest to p (found by exhaustive search).

- t -APPROXIMATE MINIMUM SPANNING TREES. Let $t > 1$, be a real number. A tree connecting the points of P is called a t -approximate minimum spanning tree of P , if its weight is at most t times the weight of the Euclidean minimum spanning tree of P . Refer to Algorithm 6. In d -space, it runs in $O(n \log n + n/(t-1)^d)$ time [6].

■ **Algorithm 6** t -APPROXIMATEMINIMUMSPANNINGTREE($P, t > 1$).

Compute the t -spanner G using Algorithm 2. Using Prim's algorithm compute a minimum spanning tree T of G . Return T .

References

- 1 Fred Anderson, Anirban Ghosh, Matthew Graham, Lucas Mougeot, and David Wisnosky. An interactive tool for experimenting with bounded-degree plane geometric spanners (media exposition). In *37th International Symposium on Computational Geometry (SoCG 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- 2 Paul B Callahan and S Rao Kosaraju. Faster algorithms for some geometric graph problems in higher dimensions. In *SODA*, volume 93, pages 291–300, 1993.
- 3 Paul B Callahan and S Rao Kosaraju. Algorithms for dynamic closest pair and n -body potential fields. In *SODA*, volume 95, pages 263–272, 1995.
- 4 Paul B Callahan and S Rao Kosaraju. A decomposition of multidimensional point sets with applications to k -nearest-neighbors and n -body potential fields. *Journal of the ACM (JACM)*, 42(1):67–90, 1995.
- 5 Sariel Har-Peled. *Geometric approximation algorithms*. Number 173 in Mathematical Surveys and Monographs. American Mathematical Soc., 2011.
- 6 Giri Narasimhan and Michiel Smid. *Geometric spanner networks*. Cambridge University Press, 2007.
- 7 Michiel Smid. The well-separated pair decomposition and its applications. In *Handbook of Approximation Algorithms and Metaheuristics*, pages 71–84. Chapman and Hall/CRC, 2018.

Subdivision Methods for Sum-Of-Distances Problems: Fermat-Weber Point, n-Ellipses and the Min-Sum Cluster Voronoi Diagram

Ioannis Mantas ✉ 

Faculty of Informatics, Università della Svizzera italiana (USI), Lugano, Switzerland

Evanthia Papadopoulou ✉ 

Faculty of Informatics, Università della Svizzera italiana (USI), Lugano, Switzerland

Martin Suderland ✉ 

Faculty of Informatics, Università della Svizzera italiana (USI), Lugano, Switzerland

Chee Yap ✉ 

Courant Institute, New York University (NYU), NY, USA

Abstract

Given a set P of n points, the *sum of distances function* of a point x is $d_P(x) := \sum_{p \in P} \|x - p\|$. Using a *subdivision approach* with *soft predicates* we implement and visualize approximate solutions for three different problems involving the sum of distances function in \mathbb{R}^2 . Namely, (1) finding the *Fermat-Weber point*, (2) constructing *n-ellipses* of a given set of points, and (3) constructing the *nearest Voronoi diagram under the sum of distances function*, given a set of point clusters as sites.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Fermat point, geometric median, Weber point, Fermat distance, sum of distances, n-ellipse, multifocal ellipse, min-sum Voronoi diagram, cluster Voronoi diagram

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.69

Category Media Exposition

Supplementary Material *Audiovisual (Video)*: <https://youtu.be/wgG8uqLIizo>

Funding *Evanthia Papadopoulou*: supported by the SNF project 200021E_201356.

Martin Suderland: supported by the SNF project 200021E_201356.

Chee Yap: supported by NSF Grant No. CCF-2008768.

1 Introduction

Let P denote a set of n points in \mathbb{R}^2 . The *sum of distances*, or *Fermat distance*, function of a point $x \in \mathbb{R}^2$ to a set P is $d_P(x) := \sum_{p \in P} \|x - p\|$, where $\|\cdot\|$ denotes the Euclidean distance. We are considering the following problems involving the Fermat distance function.

- The **Fermat(-Weber) point** of a set of points P is a point in \mathbb{R}^2 that minimizes the Fermat distance, i.e., $p_P^* := \min_{x \in \mathbb{R}^2} d_P(x)$. The *Fermat radius* is the distance realizing the Fermat point, i.e., $d_P^* := d_P(p_P^*)$. See Figure 1 (left) for an illustration.
- An **n-ellipse** of a set of n points P of *radius* r , is the level set of the Fermat distance function $d_P^{-1}(r) := \{x \in \mathbb{R}^2 \mid d_P(x) = r\}$. An n -ellipse is non-empty only if $r \geq d_P^*$. See Figure 1 (middle) for an illustration.
- The **min-sum Voronoi diagram** of a family \mathcal{S} of point sets, called *clusters*, is the subdivision of \mathbb{R}^2 into maximal regions, such that the region of a cluster $P \in \mathcal{S}$ is the locus of points closer to P than to any other cluster in \mathcal{S} , i.e., $\text{vreg}(P) := \{x \in \mathbb{R}^2 \mid d_P(x) < d_Q(x) \forall Q \in \mathcal{S} \setminus \{P\}\}$. See Figure 1 (right) for an illustration.



© Ioannis Mantas, Evanthia Papadopoulou, Martin Suderland, and Chee Yap; licensed under Creative Commons License CC-BY 4.0

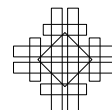
38th International Symposium on Computational Geometry (SoCG 2022).

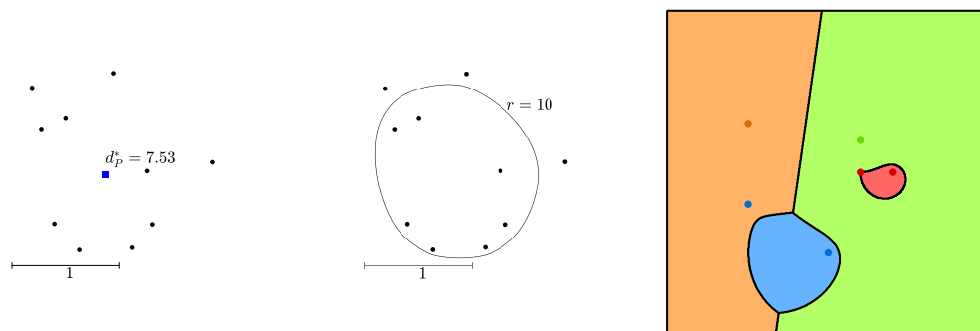
Editors: Xavier Goaoc and Michael Kerber; Article No. 69; pp. 69:1–69:6

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** Illustration of the problems considered. (left) The Fermat point (■) of 10 points. (middle) An n -ellipse of 10 points of radius 10. (right) The min-sum Voronoi diagram of 4 clusters.

Contribution. In this work we present algorithms on how to find approximate solutions to the three aforementioned problems within a starting *box* (axis-aligned rectangle), using a *subdivision approach* augmented with *soft predicates*. This box is recursively split in a quadtree fashion. Deciding whether a box should be split or not, is done with respect to some *tests*, which we perform on this box. We typically derive the tests from *predicates*, evaluated with *interval arithmetic*. In the rest of the paper, we briefly describe how our algorithms work in each of the three problems, accompanied by illustrations from our visualization tool. All algorithms directly generalize for weighted input points P .

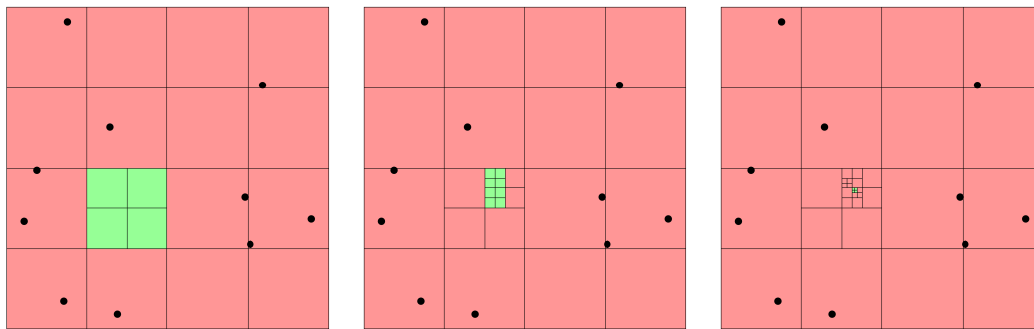
2 Problem 1: Finding the Fermat point

Finding the Fermat point (or Fermat-Weber point [25]) is an old geometric problem dating back to P. Fermat (1607–1665), which has attracted the attention of researchers of the last centuries. Unless P is a collinear point set of even size, the Fermat point is unique. Unfortunately, the coordinates of p_P^* are roots of polynomials of degree exponential in n , more precisely up to 2^n , see [5, 19]. For this reason there has been a profound interest in approximating the Fermat point; see indicatively [4, 8, 9, 10, 13, 21, 12].

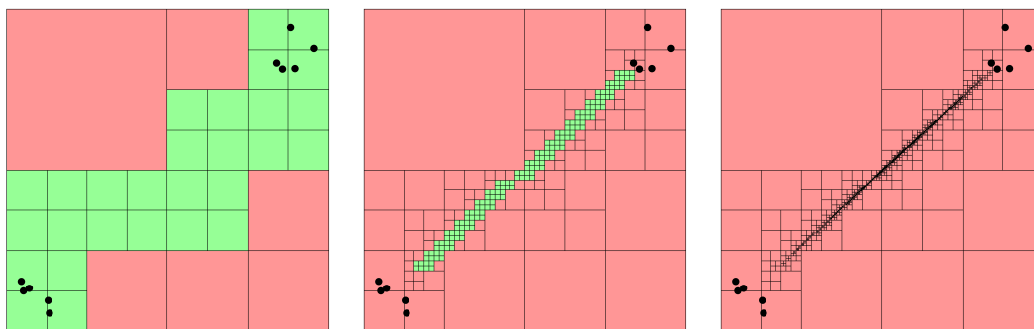
Algorithm overview. Our algorithm returns a point \widetilde{p}_F which is an ε approximation to the Fermat point, in the sense that $\|\widetilde{p}_F - p_P^*\| \leq \varepsilon$; see our paper [15] for details including improvements using Newton’s method. An illustration of the algorithm execution on two instances is shown in Figures 2 and 3. The algorithm starts with an initial box B_0 containing P , which guarantees that $p_P^* \in B_0$. During the subdivision, we keep and split boxes B that might contain p_P^* (**green boxes** in Figures 2 and 3). Boxes that are guaranteed not to contain p_P^* are discarded (**red boxes**); this is determined using an *exclusion test*. The algorithm stops when the set of remaining boxes (green) fit into a bounding box of radius ε ; this *stopping test* guarantees that the center of the bounding box is within ε distance to p_P^* .

3 Problem 2: Constructing n -ellipses

Constructing n -ellipses is also a very old geometric problem dating back to E. von Tschirnhaus (1651–1708) [24]. When $n = 1$, the curve d_P^{-1} is a circle, and when $n = 2$, it is the classic ellipse. An n -ellipse is a convex piecewise smooth curve, with singularities occurring at



■ **Figure 2** Different steps during the execution of the Fermat point algorithm (“easy” instance).



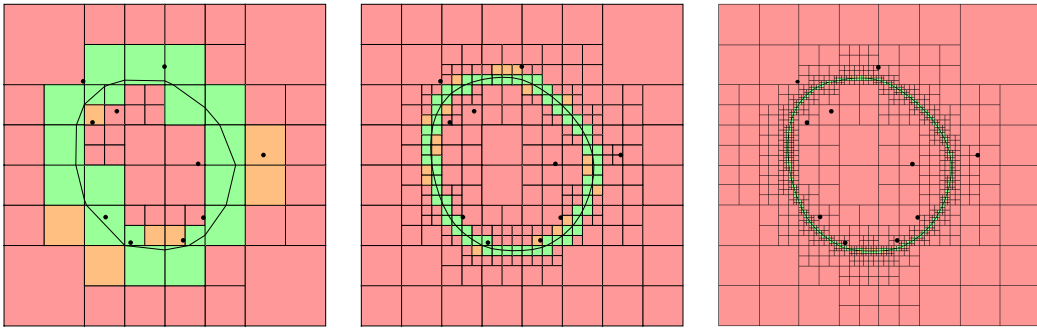
■ **Figure 3** Different steps during the execution of the Fermat point algorithm (“difficult” instance).

points of P [18, 23]. Further, analogously to the Fermat point, the polynomial equations defining the n -ellipses have algebraic degree exponential in n [19], hence there is an interest in designing approximation algorithms to construct n -ellipses.

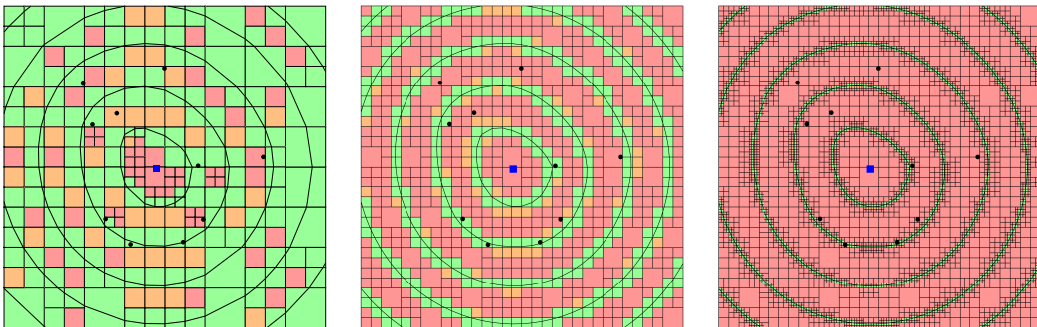
Algorithm overview. Our algorithm returns a curve E which is *isotopic* to d_P^{-1} and the Hausdorff distance between the two curves is at most ε ; refer to our paper [15] for details. An illustration of different steps of the algorithm is shown in Figure 4.

In a nutshell, the algorithm can be considered as an “*online*” *PV-construction* [16, 22]. The *PV-construction* yields isotopic approximations to a target curve, assuming that this curve is regular. The n -ellipse, though, is not regular when it passes through P [23]. During the subdivision, we keep and split boxes B until the *PV-construction* is possible in each of them; these boxes either definitely contain a piece of d_P^{-1} (**green boxes** in Figure 4) or might do so (**orange boxes**). Boxes guaranteed not to contain a piece of d_P^{-1} are discarded (**red boxes**). To ensure that E is an ε -approximation to d_P^{-1} , we split the boxes in which we draw edges until they have size ε . Boxes near P , which are additionally close to the n -ellipse (**gray boxes**), require special treatment. For each such group of gray boxes we connect the two incoming sides of the n -ellipse by just a single edge, if the group fits into a small bounding box of size ε .

Elliptic contour plotting. The described algorithm can also be used to produce isotopic ε -approximate *elliptic contour plots*, which are roughly equally spaced. By adapting the algorithm, we can simultaneously construct multiple ellipses of different radii within the same box subdivision (each ellipse corresponding to a contour line). See Figure 5 for an examples.



■ **Figure 4** Different steps during the execution of the n -ellipses algorithm.



■ **Figure 5** Different steps during the execution of the elliptic contour plotting algorithm.

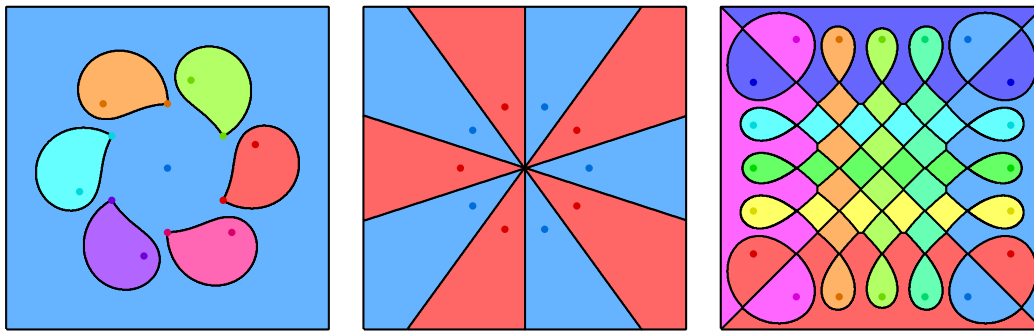
4 Problem 3: Constructing the min-sum Voronoi diagram

The min-sum Voronoi diagram of a set of point clusters is the nearest *cluster Voronoi diagram* under the Fermat distance function; refer to Figure 6 for some instances. This diagram has not been studied before, except a special case for input clusters of size 2 [6]. Various other cluster Voronoi diagrams have been considered such as the (min-max) *Hausdorff Voronoi diagram* [2, 11, 20], and the (max-min) *farthest color Voronoi diagram* [1, 14, 17].

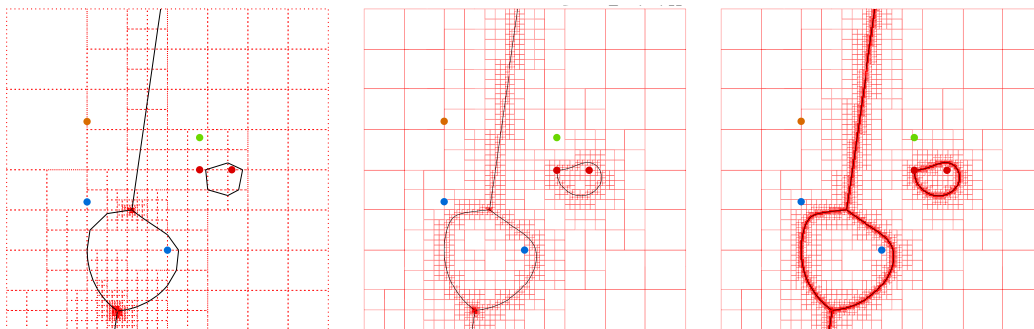
Each cluster may have a different size, in fact, the diagram can be seen as a weighted Voronoi diagram of point sites [3], where the weight of each point is determined by the cluster size. Only the clusters of the smallest size may have unbounded faces, see Figure 6(left). Further, given two clusters their bisector is smooth everywhere unless it passes through a cluster point, see Figure 6 (left).

The diagram has $\Omega(n + m^2)$ worst-case complexity, where m is the number of clusters and n is the total number of points. (1) Choose two clusters of $n/2$ points on a circle, such that the points are equally spaced and alternate between the clusters, see Figure 6 (middle). The diagram then consists of n cones emanating from the origin. (2) Choose $m = n/2$ many clusters of size 2, such that the line segments formed by connecting the 2 points of each cluster form a grid structure, see Figure 6 (right). The diagram splits into $\Omega(m^2)$ many faces.

Algorithm overview. Our algorithm returns a plane graph which is an approximation of the min-sum Voronoi diagram of \mathcal{S} with ε Hausdorff distance. It is based on a variant of the algorithm presented in [7]; refer therein for details. In brief, the edges are drawn based on the PV-construction, and in order to get an ε -approximation, prior to drawing the edges, the boxes are split until they are of size ε . Refer to Figure 7 for an illustration of the algorithm.



■ **Figure 6** Three instances of a min-sum Voronoi diagram.



■ **Figure 7** Different steps during the execution of the algorithm for min-sum Voronoi diagram.

References

- 1 Manuel Abellanas, Ferran Hurtado, Christian Icking, Rolf Klein, Elmar Langetepe, Lihong Ma, Belén Palop, and Vera Sacristán. The farthest color Voronoi diagram and related problems. In *Proceedings of the 17th European Workshop on Computational Geometry (EuroCG 2001)*, pages 113–116, 2001.
- 2 Elena Arseneva and Evanthia Papadopoulou. Randomized incremental construction for the Hausdorff Voronoi diagram revisited and extended. *Journal of Combinatorial Optimization*, 37(2):579–600, 2019.
- 3 Franz Aurenhammer and Herbert Edelsbrunner. An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern recognition*, 17(2):251–257, 1984.
- 4 Mihai Badoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pages 250–257. ACM, 2002.
- 5 Chanderjit Bajaj. The algebraic degree of geometric optimization problems. *Discrete & Computational Geometry*, 3(2):177–191, 1988.
- 6 Gill Barequet, Matthew T Dickerson, and Robert L Scot Drysdale. 2-point site Voronoi diagrams. *Discrete Applied Mathematics*, 122(1-3):37–54, 2002.
- 7 Huck Bennett, Evanthia Papadopoulou, and Chee Yap. Planar minimization diagrams via subdivision with applications to anisotropic Voronoi diagrams. *Computer Graphics Forum*, 35(5):229–247, 2016.
- 8 Prosenjit Bose, Anil Maheshwari, and Pat Morin. Fast approximations for sums of distances, clustering and the Fermat-Weber problem. *Computational Geometry*, 24(3):135–146, 2003.
- 9 Hui Han Chin, Aleksander Madry, Gary L. Miller, and Richard Peng. Runtime guarantees for regression problems. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS '13)*, pages 269–282. ACM, 2013.

- 10 Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC '16)*, pages 9–21. ACM, 2016.
- 11 Herbert Edelsbrunner, Leonidas J Guibas, and Micha Sharir. The upper envelope of piecewise linear functions: algorithms and applications. *Discrete & Computational Geometry*, 4(1):311–336, 1989.
- 12 Sándor P Fekete, Joseph SB Mitchell, and Karin Beurer. On the continuous Fermat-Weber problem. *Operations Research*, 53(1):61–76, 2005.
- 13 Sarel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007.
- 14 Daniel P Huttenlocher, Klara Kedem, and Micha Sharir. The upper envelope of Voronoi surfaces and its applications. *Discrete & Computational Geometry*, 9(3):267–291, 1993.
- 15 Kolja Junginger, Ioannis Mantas, Evanthia Papadopoulou, Martin Suderland, and Chee Yap. Certified approximation algorithms for the Fermat point and n-ellipses. In *Proceedings of the 29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- 16 Long Lin and Chee Yap. Adaptive isotopic approximation of nonsingular curves: the parameterizability and nonlocal isotopy approach. *Discrete & Computational Geometry*, 45(4):760–795, 2011.
- 17 Ioannis Mantas, Evanthia Papadopoulou, Vera Sacristán, and Rodrigo I Silveira. Farthest color Voronoi diagrams: Complexity and algorithms. In *Proceedings of the 14th Latin American Symposium on Theoretical Informatics (LATIN 2020)*, pages 283–295. Springer, 2021.
- 18 Gyula Sz Nagy. Tschirnhaus’sche Eiflächen und Eikurven. *Acta Mathematica Academiae Scientiarum Hungarica*, 1(1):36–45, 1950.
- 19 Jiawang Nie, Pablo A. Parrilo, and Bernd Sturmfels. Semidefinite representation of the k-ellipse. In *Algorithms in algebraic geometry*, pages 117–132. Springer, 2008.
- 20 Evanthia Papadopoulou and Der-Tsai Lee. The Hausdorff Voronoi diagram of polygonal objects: A divide and conquer approach. *International Journal of Computational Geometry & Applications*, 14(06):421–452, 2004.
- 21 Pablo A. Parrilo and Bernd Sturmfels. Minimizing polynomial functions. *Algorithmic and quantitative real algebraic geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 60:83–99, 2003.
- 22 Simon Plantinga and Gert Vegter. Isotopic approximation of implicit curves and surfaces. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing (SGP)*, pages 245–254, 2004.
- 23 Junpei Sekino. n-ellipses and the minimum distance sum problem. *The American mathematical monthly*, 106(3):193–202, 1999.
- 24 Ehrenfried Walther von Tschirnhaus. *Medicina Mentis Et Corporis*. Fritsch, Lipsiae, 1695. URL: <http://mdz-nbn-resolving.de/urn:nbn:de:bvb:12-bsb10008248-3>.
- 25 Alfred Weber. Über den Standort der Industrien. *Tübingen: Verlag von JCB Mohr*, 1909.

An Interactive Framework for Reconfiguration in the Sliding Square Model

Willem Sonke  

TU Eindhoven, The Netherlands

Jules Wolms  

TU Wien, Austria

Abstract

We describe SquareSlider, a software framework for visualizing reconfiguration algorithms of modular robots in the sliding square model. In this model, a robot consists of a configuration of squares in a rectangular grid, which can reconfigure through a fixed set of possible moves. SquareSlider is a web-based tool that implements an easy-to-use interface allowing the user to build a configuration, run a reconfiguration algorithm on it, and examine the results.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Modular robots, Implementation, Visualization

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.70

Category Media Exposition

Supplementary Material

Software (Web Application): <https://alga.win.tue.nl/software/squareslider/>
archived at `swh:1:dir:4970151973ae26eeaac474506f7e0fa400e414db`

Acknowledgements We thank Bettina Speckmann for her useful comments on a draft of this paper.

1 Introduction

A popular research topic in robotics and computer science is the development of modular robots [9]. These typically consist of homogeneous building blocks called *modules*, connected in such a way that the robot is able to move modules relative to each other. This way, the robot can transform by *reconfiguring* its modules. Such a reconfiguration requires careful motion planning and efficient algorithms to be viable in practice. To enable the systematic study of reconfiguration algorithms, many models of modular robots have been proposed, such as the *sliding cube model* [3, 5, 6, 7] and the *pivoting cube model* [1, 2, 4, 8].

Here, we focus on reconfiguration in the sliding cube model in two dimensions (the “*sliding square model*”). In this model, the elementary building blocks are square modules that live in the rectangular unit grid. The modules can perform two types of moves: *slides* and *convex transitions* (illustrated in Figure 1). The source and target configurations are assumed to be face-connected, and at any time during the reconfiguration, the configuration also has to stay face-connected.

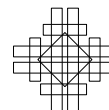
In this paper we describe a software tool called *SquareSlider* which provides a framework to visualize and interact with reconfigurations. We initially created SquareSlider to support the development of the reconfiguration algorithm Gather&Compact [3], but the framework can be useful for other algorithms as well. To this end, we designed SquareSlider to be modular: algorithm implementations are separate from the core. This way, although only Gather&Compact has currently been implemented, other algorithms can be plugged in easily.

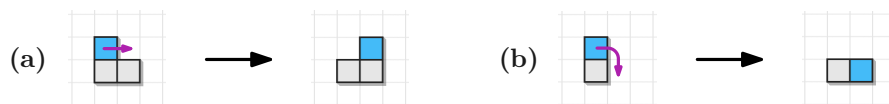
We implemented SquareSlider as a web-based tool, thus ensuring that the visualization is easily accessible, without the additional hurdle of having to compile and run a separate program. SquareSlider is available at <https://alga.win.tue.nl/software/squareslider/>.



© Willem Sonke and Jules Wolms;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 70; pp. 70:1–70:4
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



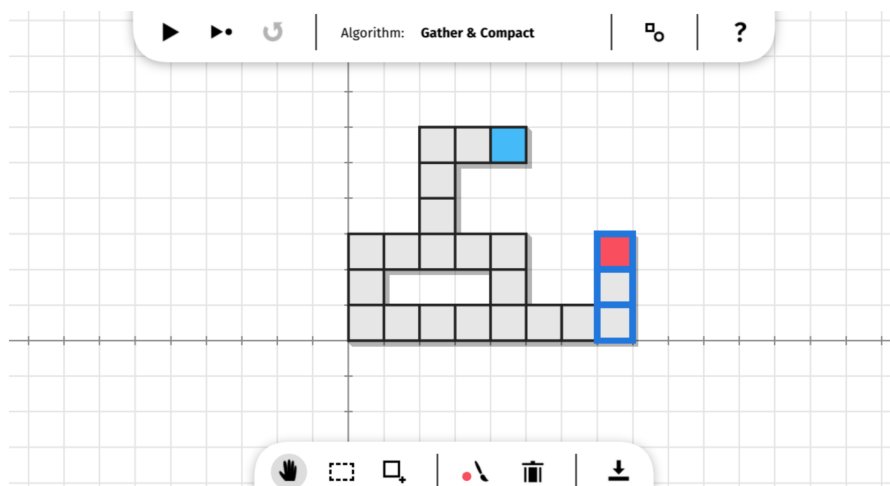


■ **Figure 1** The two types of moves in the sliding square model: (a) slide, (b) convex transition.

For the implementation, we used TypeScript,¹ a typed variant of JavaScript that compiles down to JavaScript. For the graphics, we used WebGL via the PIXI.js² project. The GUI elements used (toolbars and buttons) are custom-built. In the remainder of this paper, we will describe in detail the functionality offered by our framework.

2 Functionality

SquareSlider’s GUI consists of a large canvas that displays the configuration, and two toolbars: one on the top and one on the bottom (see Figure 2). The functionality of SquareSlider can be divided into four categories: visualization, interaction, animation, and framework utilities. We will highlight the most interesting aspects of each category.



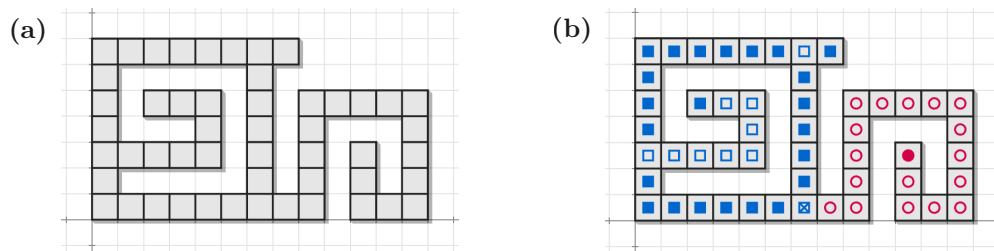
■ **Figure 2** The GUI of SquareSlider after drawing a configuration and coloring two arbitrary squares. The three squares on the right have been selected, as indicated by the blue outlines.

Visualization. By default, a configuration is visualized as gray squares in a grid, as shown in Figure 3a. The grid has an x - and y -axis defining a coordinate system, where every grid cell is indexed by its bottom-left corner. The coordinates are especially useful for interaction and animation, as will be explained below.

Many reconfiguration algorithms use information about the connectivity of the configuration, such as partitioning the squares into components, depending on their k -connectivity. We therefore augment the visualization of the squares with marks indicating this information. Currently, SquareSlider can capture the type of connectivity used by Gather&Compact [3] (see Figure 3b). Firstly, a square has a filled (blue or red) mark if it can perform moves without disconnecting the configuration, while a non-filled mark indicates that moving the square

¹ <https://www.typescriptlang.org/>

² <https://pixijs.com/>



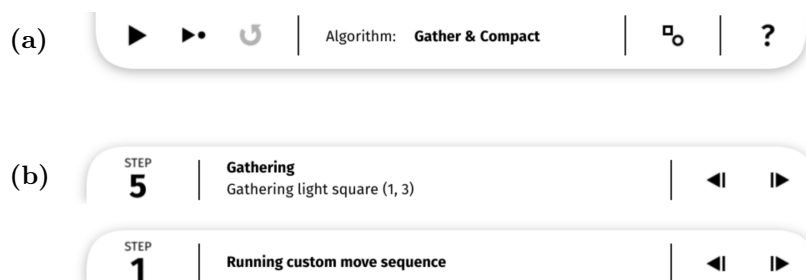
■ **Figure 3** (a) A configuration visualized in our framework. (b) The same configuration, with connectivity marks added (see the main text for an explanation of the marks).

disconnects the configuration (i.e., this square forms a cut vertex in the adjacency graph of the configuration). Secondly, the color and shape of the marks indicates a partitioning of the configuration into (roughly) 2-connected components. Blue square marks indicate squares in a *chunk*, a variant of a 2-connected component introduced in [3]. Specifically, a chunk is defined by an inclusion-maximal cycle C along with all its degree-1 neighbors and any squares inside C . Red circle marks indicate squares that are not part of a chunk. A special crossed mark shows where components connect to each other.

We use a variant of the classic algorithm to compute 2-connected components in graphs, which is based on DFS. This algorithm can easily be adapted for different notions of connectivity, as may be required by other reconfiguration algorithms.

Interaction. Input configurations can be provided to SquareSlider in two ways. Firstly, the editing tools in the bottom toolbar (see Figure 2) allow users to place and remove squares by clicking or dragging over grid cells. Clicking an empty grid cell adds a square in that cell, while clicking an existing square removes that square. Dragging over a series of cells adds or removes squares in all these cells, thus enabling quick editing of large configurations. Furthermore, users can color squares to see how a specific square or set of squares moves during reconfiguration. Secondly, users can export the configuration to a JSON string, which can later be loaded again. Such a JSON string can also be generated by an external tool.

Animation. Once a connected configuration is built, a reconfiguration can be performed. The top toolbar provides functionality to start an animation of the moves in such a reconfiguration.



■ **Figure 4** The toolbars for animating reconfigurations. (a) The top toolbar. The left two buttons allow the user to execute a reconfiguration algorithm, either by playing the whole sequence of moves, or just a single step. The circular arrow button resets the configuration to the state before reconfiguration. (b) The bottom toolbar (shown here for both Gather&Compact and a custom move sequence) showing status information about the currently running algorithm.

While the reconfiguration is running, the bottom toolbar is repurposed to show details on the performed moves (see Figure 4). Functionality is available to play the reconfiguration in one go (at various speeds), walk through the reconfiguration step by step, and to reset the configuration to the original state (after which the bottom toolbar returns to its original state, such that the configuration can be edited again).

Besides running an algorithm implementation, SquareSlider also has the option to manually input a JSON string containing a sequence of moves. This can be useful to interface with external tools. For example, the experiments in [3] used this functionality to interface with the original implementation of [7].

Framework utilities. SquareSlider provides a wide range of utility functions that can be used by algorithm implementations. For example, this includes functionality to check if a configuration is connected, to determine if a given move is valid, to enumerate all squares on the outer boundary of the configuration, and to compute a shortest sequence of moves to move a given square to a target location.

An algorithm implementation consists of a TypeScript class containing a function that produces moves that the algorithm wishes to perform. To ensure robustness, the core checks if moves performed by an algorithm are valid in the sliding square model. Any invalid moves halt the execution of the algorithm.



References

- 1 Hugo A. Akitaya, Esther M. Arkin, Mirela Damian, Erik D. Demaine, Vida Dujmović, Robin Flatland, Matias Korman, Belén Palop, Irene Parada, André van Renssen, and Vera Sacristán. Universal reconfiguration of facet-connected modular robots by pivots: The $O(1)$ musketeers. *Algorithmica*, 83(5):1316–1351, 2021. doi:10.1007/s00453-020-00784-6.
- 2 Hugo A. Akitaya, Erik D. Demaine, Andrei Gonczi, Dylan H. Hendrickson, Adam Hesterberg, Matias Korman, Oliver Korten, Jayson Lynch, Irene Parada, and Vera Sacristán. Characterizing universal reconfigurability of modular pivoting robots. In *Proc. 37th International Symposium on Computational Geometry (SoCG)*, pages 10:1–10:20, 2021. doi:10.4230/LIPIcs.SoCG.2021.10.
- 3 Hugo A. Akitaya, Erik D. Demaine, Matias Korman, Irina Kostitsyna, Irene Parada, Willem Sonke, Bettina Speckmann, Ryuhei Uehara, and Jules Wulms. Compacting squares: Input-sensitive in-place reconfiguration of sliding squares. *CoRR*, abs/2105.07997, 2021. arXiv:2105.07997.
- 4 Nora Ayanian, Paul J. White, Ádám Hálász, Mark Yim, and Vijay Kumar. Stochastic control for self-assembly of XBots. In *Proc. ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE)*, pages 1169–1176, 2008. doi:10.1115/DETC2008-49535.
- 5 Adrian Dumitrescu and János Pach. Pushing squares around. *Graphs and Combinatorics*, 22:37–50, 2006. doi:10.1007/s00373-005-0640-1.
- 6 Robert Fitch, Zack Butler, and Daniela Rus. Reconfiguration planning for heterogeneous self-reconfiguring robots. In *Proc. 2003 IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pages 2460–2467, 2003. doi:10.1109/IR0S.2003.1249239.
- 7 Joel Moreno and Vera Sacristán. Reconfiguring sliding squares in-place by flooding. In *Proc. 36th European Workshop on Computational Geometry (EuroCG)*, pages 32:1–32:7, 2020.
- 8 Cynthia Sung, James Bern, John Romanishin, and Daniela Rus. Reconfiguration planning for pivoting cube modular robots. In *Proc. 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1933–1940, 2015. doi:10.1109/ICRA.2015.7139451.
- 9 Mark Yim, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins, and Gregory S. Chirikjian. Modular self-reconfigurable robot systems. *IEEE Robotics & Automation Magazine*, 14(1):43–52, 2007. doi:10.1109/MRA.2007.339623.

Shadoks Approach to Minimum Partition into Plane Subgraphs

Loïc Crombez  

LIMOS, Université Clermont Auvergne, Aubière, France

Guilherme D. da Fonseca  

LIS, Aix-Marseille Université, France

Yan Gerard  

LIMOS, Université Clermont Auvergne, Aubière, France

Aldo Gonzalez-Lorenzo  

LIS, Aix-Marseille Université, France

Abstract

We explain the heuristics used by the **Shadoks** team to win first place in the CG:SHOP 2022 challenge that considers the minimum partition into plane subgraphs. The goal is to partition a set of segments into as few subsets as possible such that segments in the same subset do not cross each other. The challenge has given 225 instances containing between 2500 and 75000 segments. For every instance, our solution was the best among all 32 participating teams.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Plane graphs, graph coloring, intersection graph, conflict optimizer, line segments, computational geometry

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.71

Category CG Challenge

Supplementary Material *Software (Source Code)*: <https://github.com/gfonsecabr/shadoks-CGSHOP2022>; archived at [swh:1:dir:ec88e5b901c034d5a91aa133e824d65cff3788a3](https://www.swh.io/dir/ec88e5b901c034d5a91aa133e824d65cff3788a3)

Funding *Guilherme D. da Fonseca*: This work is supported by the French ANR PRC grant ADDS (ANR-19-CE48-0005).

Yan Gerard: This work is supported by the French ANR PRC grant ADDS (ANR-19-CE48-0005).

Aldo Gonzalez-Lorenzo: This work is supported by the French ANR PRC grant COHERENCE4D (ANR-20-CE10-0002).

Acknowledgements We would like to thank H el ene Toussaint, Rapha el Amato, Boris Lonjon, and William Guyot-L enat from LIMOS, as well as the Qarma and TALEP teams and Manuel Bertrand from LIS, who continue to make the computational resources of the LIMOS and LIS clusters available to our research. We would also like to thank the challenge organizers and other competitors for their time, feedback, and making this whole event possible.

1 Introduction

This paper presents our strategy to win first place in the CG:SHOP 2022 geometric optimization challenge. This edition proposed a problem called *minimum partition into plane subgraphs*. The goal is to partition the set of the edges of a given graph G embedded in the plane (with line segments as edges) into a small number k of plane graphs. The problem reduces to graph coloring a conflict graph G_C where the vertices of G_C are the segments of G and two vertices of G_C are connected by an edge if the corresponding segments cross each other (for details on the definition of *cross*, see [4]).



  Lo c Crombez, Guilherme D. da Fonseca, Yan Gerard, and Aldo Gonzalez-Lorenzo;
licensed under Creative Commons License CC-BY 4.0

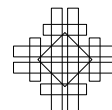
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 71; pp. 71:1–71:8

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum f ur Informatik, Dagstuhl Publishing, Germany



The study of graph coloring goes back to the 4-color problem (1852) and the problem has been intensively studied since the 1970s [9]. Many heuristics have been proposed [6, 8, 13, 14], as well as exact algorithms [3, 7, 12] (see for instance the book [11]). In this paper we present the ideas we used in the competition. The main element is a *Conflict Optimizer*, that does not use any geometry. It is based on the same approach we used to solve low-makespan coordinated motion planning in the CG:SHOP 2021 challenge [2]. Our initial solutions, however, make extensive use of geometry. The code is available on github.

The paper is organized as follows. Section 2 presents some heuristics that we used to compute initial solutions. In Section 3 we describe the technique used to improve a solution. Section 4 details our implementation of the algorithm and a parameter analysis. Section 5 describes the results we obtained.

2 Initial Solutions

The most simple method to produce a (reasonably small) coloring of a graph is the classic greedy heuristic: for each segment e , we color e using the first color *available*, i.e. that is not already used by any of the segments that cross e . If necessary, we create a new color. The order by which we consider the segments influences the quality of the solution. We refer to the greedy heuristic using the order by which the segments appear on the instance files as **Greedy** when comparing the results. Sorting the segments in angular order (and trying different starting angles) produces good solutions to the challenge instances. We refer to this simple heuristic as **Angle**.

The *squeaky wheel* paradigm has been widely applied to graph coloring [10]. The idea is to run a heuristic, detect elements that have been solved poorly, and run the same heuristic again handling these elements earlier this time. The procedure is repeated several times and the best solution found is returned. We use this paradigm together with **Angle** as follows. Throughout the algorithm, the segments are partitioned into two lists *Good*, *Bad*, both kept sorted by angle. Initially, all segments are in *Good*. At each step we apply the greedy coloring first treating the segments in *Bad* and then in *Good*. Then, the segments that have been assigned the last color are added to *Bad* and we repeat the procedure. The number of colors used may eventually increase (since both lists are kept sorted by angle). We stop after a certain time or number of steps and return the solution with the smallest number of colors found. We refer to this heuristic as **Bad**.

A classic variation of the greedy coloring is the **DSatur** heuristic [1]. It does not use any geometric information. At each step, we color the segment e that crosses the largest number of different colors, breaking ties by the total number of segments that cross e . As in the standard greedy heuristic, the color assigned to e is the first color that is available.

We modify the **DSatur** heuristic into the **DSHull** heuristic that uses geometric information. We color the segments following the same order criterion as **DSatur**. However, instead of assigning to e the first color available, we choose the color as follows. The segments that have the same color are kept in a set called a *color class*. For each color class C , let $w(C)$ be the area of the convex hull of the segments in C . When coloring a segment e , we choose, among the color classes C that are available for e , the one that minimizes $w(C \cup \{e\}) - w(C)$. Ties are broken arbitrarily and if no color class is available for e , then we create a new color class containing only e . The intuition is that a small increase in the convex hull areas corresponds to a compact packing of the segments, producing larger color classes.

A comparison of the heuristics on several challenge instances is presented in Table 1.

■ **Table 1** Initial solutions produced by several heuristics compared against the best solution found.

instance	density	Greedy	Angle	Bad	DSatur	DSHull	Best
rsqrpecn8051	41%	342	205	203	213	201	175
vispecn13806	19%	427	308	300	289	283	218
rsqrp14364	50%	294	139	139	165	157	136
vispecn19370	13%	370	285	278	265	248	192
visp26405	7%	154	101	97	94	92	81
visp31334	5%	152	90	88	99	98	81
visp38574	14%	287	148	146	168	168	133
sqrpecn45700	47%	952	504	500	562	522	462
reecn51526	24%	642	361	359	388	360	310
vispecn58391	12%	789	607	594	499	494	367
vispecn65831	12%	916	647	637	578	564	439
sqrp72075	47%	609	280	280	363	337	269

3 Improving Solutions

In this section we describe our optimization approach that we call *Conflict Optimizer*. Section 3.1 describes the backbone of the *conflict optimizer*. Section 3.2 describes some improvements that were made to the conflict optimizer in order to get better solutions.

3.1 Conflict optimizer

The goal of the conflict optimizer is to remove one color from a given solution with k colors. Let C_0 be a color class. The conflict optimizer puts all segments of C_0 in a queue Q and deletes C_0 . We now have a partial solution with $k - 1$ colors and a queue Q that contains uncolored segments. The goal is to empty Q by coloring every segment in Q .

At each step until Q is empty, we pop a segment e from Q and color e as follows. If there exists a color class C such that no segment in C crosses e , then we add e to C . In most cases, such C does not exist and we choose C to minimize the following cost function. Let $q(e)$ be the number of times the segment e has been added to Q . The *penalty* for adding e to Q is $1 + q(e)^p$. The *cost* of each color class C is the product of a Gaussian random variable of mean 1 and variance σ with the sum of the penalties of the segments of C that cross e . The values of the parameters p, σ are analysed in Section 4 ($p = 1.2$ and $\sigma = 0.15$ are good default values).

3.2 Modifications to the conflict optimizer

In this section we describe several modifications that we made to the conflict optimizer described in Section 3.1. In our code, we developed several options that can be toggled on or off. The impact on the computation of solutions is discussed in Section 4.

Easy segments. Given an objective number of colors k , we call *easy segments* a list of segments S such that, if the remainder of the segments of S are colored using k colors, then we are guaranteed to be able to color all segments with k colors. To obtain S we iteratively remove from the graph a segment e that has at most $k - 1$ crossings, appending e to S . We repeat until no other segment can be added to S . Notice that, once we color the remainder

of the graph with at least k colors, we can use a greedy coloring for S in order from last to first without increasing the number of colors used. Removing the easy segments reduces the total number of segments, making the conflict optimizer more effective.

Clique segments. A *clique* is a set of mutually crossing segments. We used several heuristics to produce large cliques. Let K be the largest clique we found for a given instance. Since the segments of K must have different colors, we forbid the segments in K from entering the queue by setting a infinite penalty.

Restarting. We implemented a strategy to restart the conflict optimizer. We set a hard limit q_{\max} to how many times a segment can be queued. Once a segment e has been queued q_{\max} times, the penalty of e becomes infinite. Once it becomes impossible to color a segment from the queue (that is, the minimum cost is infinite), the conflict optimizer aborts and restarts. When restarting, the coloring is shuffled by moving segments that fit multiple color classes.

Bounded Depth-First Search. The *bounded depth-first search* (BDFS) algorithm tries to improve the dequeuing process. The goal is to prevent a segment from being queued by locally recoloring a bounded number of segments in the current partial solution. To do so, we perform a local search into the tree of possible ways to color the segments.

The BDFS algorithm has two parameters: *crossing bound* c_{\max} and *depth* d . In order to recolor a segment e , BDFS gets the set \mathcal{C} of color classes with at most c_{\max} crossings with e . If a class of \mathcal{C} has no crossings with e , we assign e to C . Otherwise, for each class $C \in \mathcal{C}$, BDFS tries to recolor the list of segments in C that cross e by recursively calling itself with depth $d - 1$. At depth $d = 0$ the algorithm stops trying coloring the segments.

During the challenge we used BDFS with parameters $c_{\max} = 3$ and $d = 3$. The depth was increased to 5 (resp. 7) when the number of segments in the queue was 2 (resp. 1).

4 Implementation and Experiments

In this section, we describe the techniques we used to efficiently implement the conflict optimizer. We also analyze the influence of the different parameters and options.

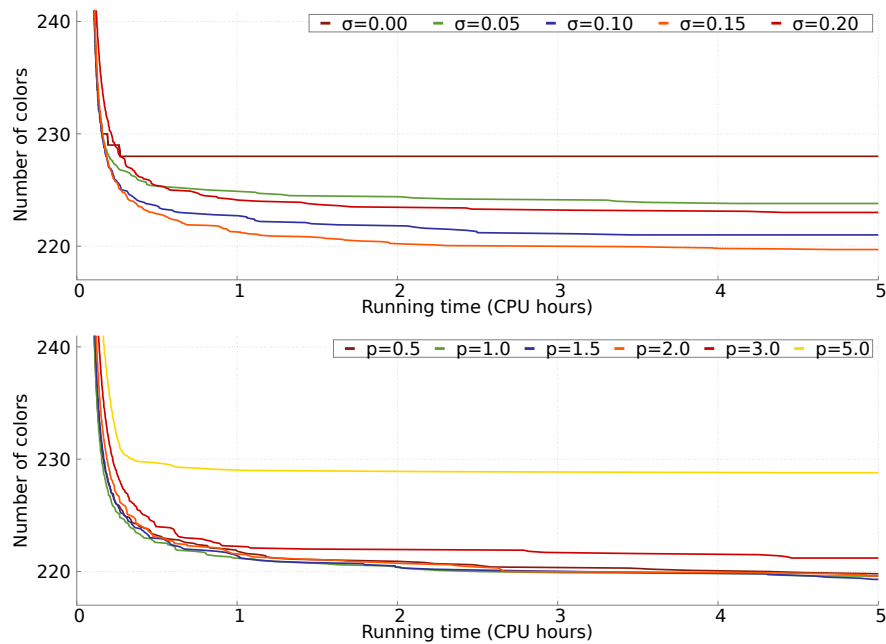
4.1 Implementation

We implemented our algorithm in C++ using only the standard library. As the conflict optimizer spends most of its time testing crossings, we precompute the crossings. To save memory space, we stored the crossing state of each pair of segments using just one bit, which allows us to store the largest instances of the challenge on less than 800MB.

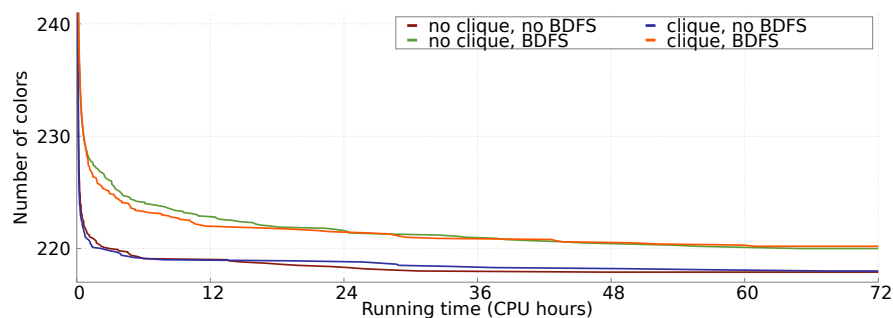
4.2 Parameter analysis

The two parameters of the conflict optimizer are the variance σ of the Gaussian noise and the exponent p of the penalty. The two others options BDFS and multistart can be activated to improve solutions that have already been optimized several times.

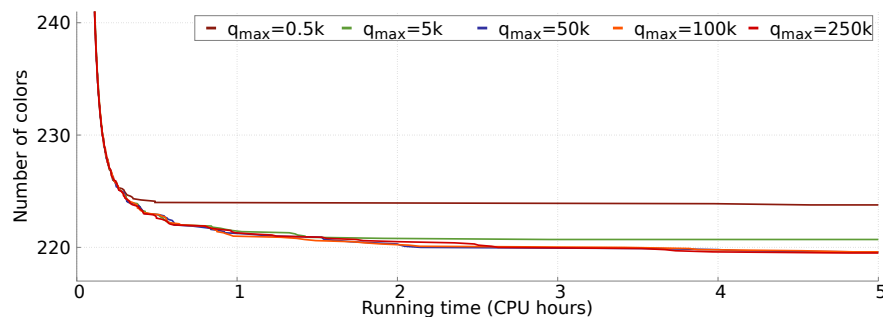
Parameters σ and p . Figure 1 shows the influence of both these parameters (the initial solutions used for the figure are computed using **Greedy**). In all figures, the number of colors shown is the average of multiple executions of the code using different random numbers.



■ **Figure 1** Number of colors over time for the instance *vispecn13806* using different parameters. In both figures the algorithm uses easy segments, $q_{\max} = 59022$, but does not use the BDFS nor any clique. The first plot shows results with different values of σ for $p = 1.2$. The second plot shows results with different values of p for $\sigma = 0.15$.



■ **Figure 2** Number of colors over time with and without clique knowledge and BDFS obtained on the instance *vispecn13806*. Parameters are $\sigma = 0.15$, $p = 1.2$, and $q_{\max} = 1500000$.



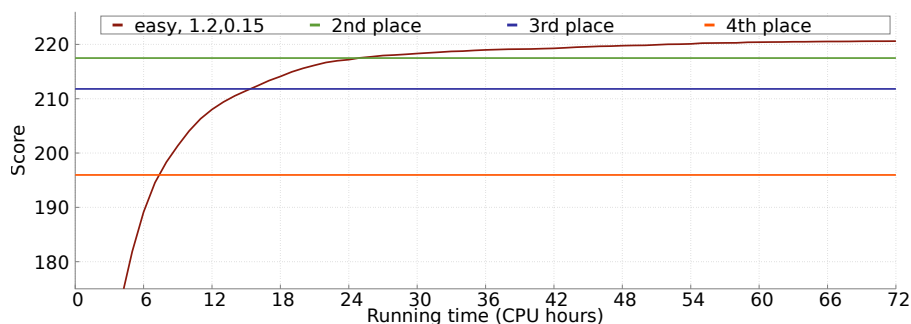
■ **Figure 3** Number of colors over time with different values of q_{\max} obtained on the instance *vispecn13806*. Parameters are $\sigma = 0.15$, $p = 1.2$, no clique knowledge, and no BDFS.

Options multistart and BDFS. The goal of multistart and BDFS is to further optimize very good solutions that the conflict optimizer is not able to improve otherwise. Figure 2 shows the influence of large clique knowledge and BDFS. While on this figure, the advantages of BDFS cannot be noticed, its use near the end of the challenge improved about 30 solutions.

Looking at Figure 3, the maximal number of times a segment can be queued does not seem to have much influence as long as its value is not too small. Throughout the challenge we almost exclusively used $q_{\max} = 2000 \cdot (75000/m)^2$, where m is the number of segments. This value roughly ensures a restart every few hours.

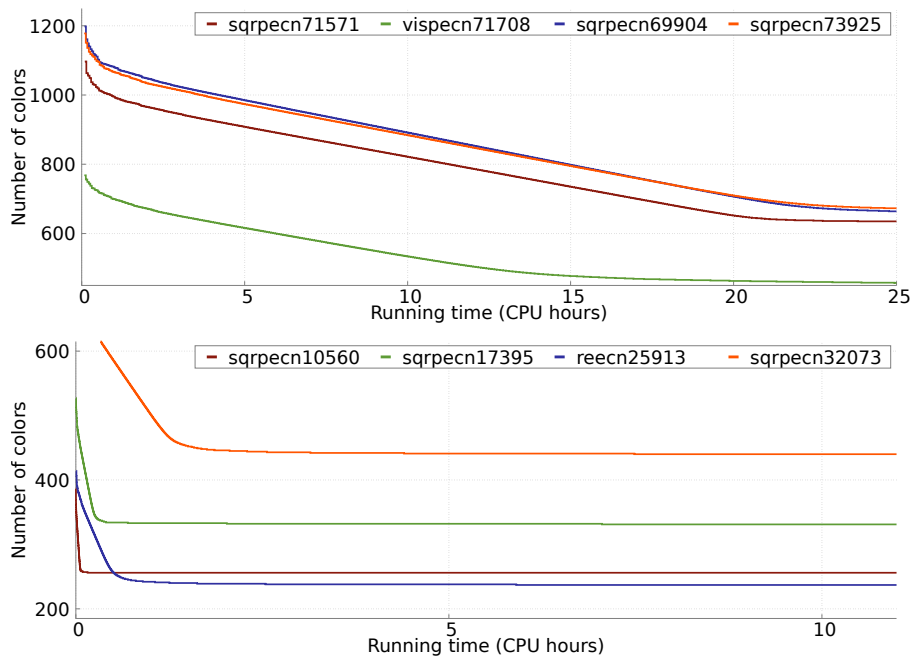
5 Challenge Results

We won first place in the challenge with the best solution among all 32 participating teams for all 225 instances. We also showed that 23 of those solutions are optimal by identifying a clique as large as the number of colors.



■ **Figure 4** Evolution of the score over time compared to the scores of second to fourth place. The same parameters are used on all instances ($p = 1.2, \sigma = 0.15$, and easy segments are computed).

After generating initial solutions we ran our conflict optimizer with various parameters. The clique knowledge and the easy segments reduction were always used. Most of the time we used $\sigma = 0.15 \pm 0.05$ and $p = 1.2 \pm 0.1$. The BDFS strategy was used in the last couple of weeks of the challenge. We estimate that on average, we spent two to three weeks of single core of an Intel Xeon E5-2670 CPU per instance. However, despite the large amount of computing power used during the challenge, and the varying parameters of our algorithms, we note that after 25 hours of computation on each file, starting from the Greedy solution, using only the easy segments optimization and parameters $p = 1.2, \sigma = 0.15$, our conflict optimizer reaches a score of 217.64 on the CG:SHOP 2022 instances, which is better than the second place score (see Figures 4, 5). We note that the second and third team [16, 5] also use a conflict optimizer heuristic, while the fourth team [15] uses instead a SAT solver coupled with tabu search. Despite several parameters that allow for increased diversity in order to find really good solutions, our conflict optimizer still performs well with default parameters. Finally, as the optimizer does not make use of any geometric property, it might be interesting in the future to test its performance on other classes of graphs.



■ **Figure 5** Challenge scores over time for several instances.



References

- 1 Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.
- 2 Loïc Crombez, Guilherme D. da Fonseca, Yan Gerard, Aldo Gonzalez-Lorenzo, Pascal Lafourcade, and Luc Libralesso. Shadoks approach to low-makespan coordinated motion planning (CG challenge). In *37th International Symposium on Computational Geometry, SoCG 2021*, pages 63:1–63:9, 2021.
- 3 David Eppstein. Small maximal independent sets and faster exact graph coloring. *J. Graph Algorithms Appl*, 7(2):131–140, 2002.
- 4 Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Stefan Schirra. Minimum partition into plane subgraphs: The CG: SHOP Challenge 2022. *CoRR*, abs/2203.07444, 2022. [arXiv: 2203.07444](https://arxiv.org/abs/2203.07444).
- 5 Florian Fontan, Pascal Lafourcade, Luc Libralesso, and Benjamin Momège. Local search with weighting schemes for the CG:SHOP 2022 competition. In *Symposium on Computational Geometry (SoCG)*, pages 73:1–73:7, 2022.
- 6 Philippe Galinier and Jin-Kao Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of combinatorial optimization*, 3(4):379–397, 1999.
- 7 Stefano Gualandi and Federico Malucelli. Exact solution of graph coloring problems via constraint programming and column generation. *INFORMS Journal on Computing*, 24(1):81–100, 2012.
- 8 Alain Hertz and Dominique de Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, 1987.
- 9 Tommy R. Jensen and Bjarne Toft. *Graph coloring problems*. John Wiley & Sons, 2011.
- 10 David E. Joslin and David P. Clements. Squeaky wheel optimization. *Journal of Artificial Intelligence Research*, 10:353–373, 1999.
- 11 R. M. R. Lewis. *A Guide to Graph Colouring: Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2015.



71:8 Shadoks Approach to Minimum Partition into Plane Subgraphs

- 12 Corinne Lucet, Florence Mendes, and Aziz Moukrim. An exact method for graph coloring. *Computers & Operations Research*, 33(8):2189–2207, 2006.
- 13 David W. Matula, George Marble, and Joel D. Isaacson. Graph coloring algorithms. In *Graph theory and computing*, pages 109–122. Elsevier, 1972.
- 14 Isabel Méndez-Díaz and Paula Zabala. A branch-and-cut algorithm for graph coloring. *Discrete Applied Mathematics*, 154(5):826–847, 2006.
- 15 André Schidler. SAT-based local search for plane subgraph partitions. In *Symposium on Computational Geometry (SoCG)*, pages 74:1–74:8, 2022.
- 16 Jack Spalding-Jamieson, Brandon Zhang, and Da Wei Zheng. Conflict-based local search for minimum partition into plane subgraphs. In *Symposium on Computational Geometry (SoCG)*, pages 72:1–72:6, 2022.

Conflict-Based Local Search for Minimum Partition into Plane Subgraphs

Jack Spalding-Jamieson  

David R. Cheriton School of Computer Science, University of Waterloo, Canada

Brandon Zhang  

Vancouver, Canada

Da Wei Zheng   

Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA

Abstract

This paper examines the approach taken by team gitastrophe in the CG:SHOP 2022 challenge. The challenge was to partition the edges of a geometric graph, with vertices represented by points in the plane and edges as straight lines, into the minimum number of planar subgraphs. We used a simple variation of a conflict optimizer strategy used by team Shadoks in the previous year's CG:SHOP to rank second in the challenge.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Design and analysis of algorithms

Keywords and phrases local search, planar graph, graph colouring, geometric graph, conflict optimizer

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.72

Category CG Challenge

Supplementary Material

Software (Source Code): <https://github.com/jacketsj/cgshop2022-gitastrophe>
archived at `swh:1:dir:0e86e287cc9a882064e46283cb35cbd64b0df4e8`

1 Introduction

Given a graph $G = (V, E)$ and an assignment $f : V \rightarrow \mathbb{Z}^2$ inducing a straight-line drawing in \mathbb{R}^2 with integer vertex coordinates, the *minimum partition into plane subgraphs* problem asks for a partition of the edges E into a minimal number of sets E_1, E_2, \dots, E_k such that for each subgraph $G_i = (V, E_i)$, f induces a planar straight-line drawing. That is, no pair of edges from the same subset intersect, except possibly at their common endpoint. This was the problem posed in the 2022 Computational Geometry Challenge (CG:SHOP 2022). For more detail about the challenge, we refer readers to the summary paper [5].

Reduction to vertex-colouring

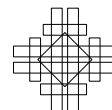
Solving the minimum partition into plane subgraphs problem for $G = (V, E)$ is equivalent to solving the well-studied minimum vertex-colouring problem for the intersection *conflict graph* G' with $V(G') = E(G)$ and $E(G')$ equal to the set of intersections in the provided straight-line drawing. We did not explicitly use the geometric properties of the instances and instead solved the aforementioned vertex colouring problem.

Henceforth, we will only refer to the intersection conflict graph G' induced by the instance. Vertices will refer to the vertices $V(G')$, and edges will refer to the edges $E(G')$. Our goal is to partition the vertices using a minimum set of colour classes $\mathcal{C} = \{C_i\}$, where no two vertices in the same colour class C_i are incident to a common edge.



© Jack Spalding-Jamieson, Brandon Zhang, and Da Wei Zheng;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 72; pp. 72:1–72:6
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Existing literature

There are many existing practical heuristic algorithms [11, 10, 13, 14, 1] to the vertex-colouring problem. Many of these algorithms used DIMACS benchmark [9] graphs to evaluate their results. In subsection 3.3 we compare the results of our methods for these instances. Most of the benchmark instances had comparatively few edges (on the order of thousands or millions); the largest intersection graphs considered in the CG:SHOP challenge had over 1.5 billion edges.

We found a variation of the *conflict optimizer* strategy employed by team Shadoks for CG:SHOP 2021 [4] to be effective. We describe this strategy in Section 2. Using this strategy, we, team *gitastrophe*, placed second overall, and first among all junior teams. This result was surprising to us, as our methods were relatively simple, relying exclusively on the naive reduction to vertex-colouring. The first- and third-place teams also make use of similar techniques [3] [6], although the fourth place team uses a very different SAT-based approach [12].

2 Methods

2.1 Solution initialization

We used the traditional greedy algorithm of Welsh and Powell [15] to obtain initial solutions: order the vertices in decreasing order of degree, and assign each vertex the minimum-label colour not used by its neighbours. We attempted to use different orderings for the greedy algorithm, such as sorting by the slope of the line segment associated with each vertex, and we also tried numerous other strategies. Ultimately, we found that after running our solution optimizer for approximately the same amount of time, all initializations resulted in equal number of colours.

2.2 Solution optimization: conflict search

Our most successful method for improvement of the solutions was inspired by the conflict optimization approach used by the Shadoks team for CG:SHOP 2021 [4]. At a high-level, our algorithm will iteratively attempt to eliminate a selected colour class. The details are as follows:

1. Pick a random colour class C to be eliminated. Uncolour all vertices in C and add all vertices in that colour class to a conflict set S . We maintain only a valid vertex-colouring for the set $V(G') - S$. Once S is empty, we will have produced a valid vertex colouring of G' which uses one fewer colour.
2. Pick and remove a random element v from S . For each colour class, we compute the *conflict score* with v . The conflict score of a colour class C_i is

$$\sum_{\substack{u \in C_i \\ (u,v) \in E(G')}} 1 + q(u)^2 \tag{1}$$

where $q(u)$ is the number of times that u has been removed from the conflict set S in previous iterations of this step.

3. Pick the colour class C_i with the lowest conflict score. Uncolour all vertices in C_i which are adjacent to v and add those vertices to S . Insert v into C_i .
4. Repeat steps 2 and 3 until the set S is empty.

There is no guarantee that this algorithm terminates. In practice, we restart the procedure when any value of $q(u)$ surpasses a fixed threshold.

The primary differences between our approach to conflict optimization and those of the first and third place teams are the choice of an exponent of 2 in Step 2, and the behaviour when $q(u)$ surpasses its fixed threshold.

Modifications to the conflict optimizer

Taking inspiration from memetic algorithms, which alternate between an intensification and a diversification stage, we continually switched between a phase where we used the above conflict score, and one where we minimized only the number of conflicts (i.e. we replaced the conflict score of (1) with $\sum_{u \in C_i, (u,v) \in E(G')} 1$). Each phase lasted for 10^5 iterations. Adding the conflict-minimization phase gave minor improvements to some of the challenge instances.

2.3 Failed approach: memetic algorithms

Although many of the leading approaches to vertex colouring are memetic, our attempts at implementing them performed poorly. These memetic algorithms take a long time to run on the standard DIMACS instances [9], and did not scale well to the much larger intersection graphs in the challenge.

We implemented the memetic algorithms Evo-Div [11] and HEAD [10], but neither of these approaches were able to improve on the scores obtained by the conflict optimizer. Both of these algorithms use TABUCOL [8], a tabu search algorithm, as their local search component, so we tried to replace it with the conflict optimizer. However, this proved to be ineffective. This may be attributed to a critical difference between TABUCOL and the conflict optimizer: the conflict optimizer does not expressly minimize the number of conflicting edges in the colouring, and only hopes to eventually resolve all conflicting vertices.

3 Results

3.1 Implementation

The conflict optimizer frequently looked up edges in the intersection graph. To speed this process up, we precomputed the adjacency matrix of the graph and stored it in memory for fast access. Our C++ implementation is available on Github.

3.2 Challenge computing environment

To perform our computations during the challenge, we mainly used a 32-core server with two Xeon E5-2698 v3s. We spent about 2 days of CPU time per instance to obtain our best solutions. Table 1 shows the scores of our greedy initialization, scores after running the conflict optimizer for 10 minutes, 1 hour, and 24 hours, and the best result we obtained in the challenge. Our algorithm obtains good results on many instances after a short period of time; it comes close to matching the best solutions we obtained in the challenge within 24 hours (and surpasses some, as there is randomness in the algorithm).

■ **Table 1** Results of our algorithm on a subset of the challenge instances after fixed amounts of optimization time. Note that on instances `visp31334` and `reecn51526` we obtained better results after 24 hours than our final results from the challenge.

Instance	Greedy	10m	1h	24h	Final
<code>rvisp5013</code>	71	50	49	49	49
<code>rsqrpecn8051</code>	284	177	176	176	176
<code>sqrp10642</code>	186	124	124	124	124
<code>rsqrp14364</code>	225	137	137	137	137
<code>reecn16388</code>	210	152	152	151	151
<code>vispecn19370</code>	285	199	196	194	194
<code>sqrpecn23715</code>	657	436	425	423	423
<code>visp26405</code>	119	83	83	82	81
<code>sqrp28863</code>	316	209	192	191	191
<code>visp31334</code>	132	83	83	83	82
<code>vispecn35198</code>	379	262	246	242	243
<code>visp38574</code>	193	143	136	135	134
<code>sqrp41955</code>	362	236	214	204	204
<code>sqrpecn45700</code>	802	503	471	465	465
<code>visp48558</code>	230	159	147	144	144
<code>reecn51526</code>	456	334	317	311	312
<code>visp55158</code>	182	130	123	122	122
<code>vispecn58391</code>	609	440	394	370	369
<code>visp62685</code>	174	132	120	119	117
<code>vispecn65831</code>	711	522	473	442	440
<code>sqrpecn69904</code>	1152	740	693	651	650
<code>sqrp72075</code>	483	342	312	272	271

3.3 Comparison on DIMACS dataset

We ran our algorithm on the difficult DIMACS instances [9] to gauge our algorithm’s performance on non-geometric graphs.

Table 2 shows our results after running our algorithm for 10 minutes, compared with some of the state of the art colouring algorithms HEAD [10] and QACOL [13, 14].

Surprisingly, the conflict optimizer works extremely poorly on random graphs, but is fast and appears to perform well on geometric graphs, matching the best-known results [7]. Interestingly, these geometric graphs are not intersection graphs as in the Challenge, but are generated based on a distance threshold.

Applying Cheeger’s inequality [2], we note the intersection graphs resulting from the challenge instances have noticeably lower edge conductance than random graphs, and we believe this plays a part in the performance of the conflict optimizer.

4 Conclusion

The conflict optimizer approach was very effective for the large geometric intersection graphs for the CG:SHOP 2022 challenge. Further investigation is needed into the reason the conflict optimizer approach was effective.

■ **Table 2** Comparison of our method with state-of-the-art graph colouring algorithms. The conflict optimizer underperforms except on the geometric graphs $rX.Y$ and $dsjrX.Y$.

Instance	Colours	HEAD [10]	QACOL [13, 14]
dsjc250.5	29	28	28
dsjc500.1	13	12	12
dsjc500.5	52	47	48
dsjc500.9	130	126	126
dsjc1000.1	21	20	20
dsjc1000.5	93	82	82
dsjc1000.9	235	222	222
r250.5	65	65	65
r1000.1c	98	98	98
r1000.5	234	245	238
dsjr500.1c	85	85	85
dsjr500.5	122	-	122
le450_25c	26	25	25
le450_25d	26	25	25
flat300_28_0	33	31	31
flat1000_50_0	91	50	-
flat1000_60_0	93	60	-
flat1000_76_0	92	81	81
C2000.5	173	146	145
C4000.5	317	266	259

References

- 1 Daniel Bréaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.
- 2 Jeff Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. *Problems in analysis*, 625(195-199):110, 1970.
- 3 Loïc Crombez, Guilherme D. da Fonseca, Yan Gerard, and Aldo Gonzalez-Lorenzo. Shadoks approach to minimum partition into plane subgraphs. In *Symposium on Computational Geometry (SoCG)*, pages 71:1–71:8, 2022.
- 4 Loïc Crombez, Guilherme D da Fonseca, Yan Gerard, Aldo Gonzalez-Lorenzo, Pascal Lafourcade, and Luc Libralesso. Shadoks approach to low-makespan coordinated motion planning (cg challenge). In *37th International Symposium on Computational Geometry (SoCG 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- 5 Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Stefan Schirra. Minimum partition into plane subgraphs: The CG: SHOP Challenge 2022. *CoRR*, abs/2203.07444, 2022. [arXiv: 2203.07444](#).
- 6 Florian Fontan, Pascal Lafourcade, Luc Libralesso, and Benjamin Momège. Local search with weighting schemes for the CG:SHOP 2022 competition. In *Symposium on Computational Geometry (SoCG)*, pages 73:1–73:6, 2022.
- 7 Olivier Goudet, Cyril Grelier, and Jin-Kao Hao. A deep learning guided memetic framework for graph coloring problems, 2021. [arXiv:2109.05948](#).
- 8 Alain Hertz and Dominique de Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, 1987.
- 9 David S Johnson and Michael A Trick. *Cliques, coloring, and satisfiability: second DIMACS implementation challenge, October 11-13, 1993*, volume 26. American Mathematical Soc., 1996.

- 10 Laurent Moalic and Alexandre Gondran. Variations on memetic algorithms for graph coloring problems. *Journal of Heuristics*, 24(1):1–24, 2018.
- 11 Daniel Cosmin Porumbel, Jin-Kao Hao, and Pascale Kuntz. An evolutionary approach with diversity guarantee and well-informed grouping recombination for graph coloring. *Computers & Operations Research*, 37(10):1822–1832, 2010.
- 12 André Schidler. SAT-based local search for plane subgraph partitions. In *Symposium on Computational Geometry (SoCG)*, pages 74:1–74:8, 2022.
- 13 Olawale Titiloye and Alan Crispin. Quantum annealing of the graph coloring problem. *Discret. Optim.*, 8:376–384, 2011.
- 14 Olawale Titiloye and Alan Crispin. Parameter tuning patterns for random graph coloring with quantum annealing. *PloS one*, 7(11):e50060, 2012.
- 15 D. J. A. Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86, January 1967.

Local Search with Weighting Schemes for the CG:SHOP 2022 Competition

Florian Fontan 

Independent Researcher, Paris, France

Pascal Lafourcade   

Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, LIMOS,
63000 Clermont-Ferrand, France

Luc Libralesso   

Atoptima, 16 Place Sainte Eulalie, 33000 Bordeaux, France

Benjamin Momège 

Independent Researcher, Clermont-Ferrand, France

Abstract

This paper describes the heuristics used by the LASAOF00FUBESTINRRALLDECA¹ team for the CG:SHOP 2022 challenge. We introduce a new greedy algorithm that exploits information about the challenge instances, and hybridize two classical local-search schemes with weighting schemes. We found 211/225 best-known solutions. Hence, with the algorithms presented in this article, our team was able to reach the 3rd place of the challenge, among 40 participating teams.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases heuristics, vertex coloring, digital geometry

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.73

Category CG Challenge

Supplementary Material *Software (Source Code)*: <https://github.com/librallu/dogs-color>
archived at `swh:1:dir:9388a1f6c982c53a827264e5503824a4ee44c224`

Funding *Pascal Lafourcade*: This work is supported by the French ANR PRC grant MobiS5 (ANR-18-CE39-0019), DECRYPT (ANR-18-CE39-0007), and SEVERITAS (ANR-20-CE39-0005).

1 Introduction

The Fourth Geometric Optimization Challenge proposed in SoCG 2022 is about finding Minimum Plane Partitions. Given a geometric graph with vertices represented by points in the plane, and edges by straight-line connections between vertices, we aim to partition edges into as few subsets of disjoint lines, and to minimize the partition size.

Solving an instance of this challenge is equivalent to solving an instance of the classical vertex coloring problem (VCP). Indeed, we can define a graph $G' = (V', E')$ where V' is the set of lines, and there is an edge in E' between two intersecting lines. Finding a valid vertex coloring of G' with k colors is equivalent to finding a coloring of the original problem with k colors. We refer the reader to the competition survey for more details about the vertex coloring problem transformation [4], and to the other participants articles [2, 15, 14].

¹ At the beginning of the competition, we could not find some team name. Thus, we used a simple permutation of the first team members names letters, hence the funny name.



In our approach for the SoCG challenge, we made the choice to focus only on solving the equivalent vertex coloring problem instances instead of directly solving the competition instances. We show that “classical” graph coloring algorithms are competitive, even with ad-hoc approaches for the SoCG challenge.

We combined some classical algorithms found in the vertex coloring literature (two local-search neighborhoods presented below) with a dynamic weighting scheme present in many methods for solving other graphs problems (and closely related to the conflict optimizer used in the previous CG:SHOP competition [3]). Furthermore, we show that this new algorithm is simple to implement, yet efficient as it enabled us to reach the 3rd place in the CG:SHOP 2022 competition.

The main idea of our approach is to optimize some solutions with some local search algorithms. We use two techniques to find some solutions: a segment orientation greedy ad-hoc algorithm, and DSATUR, one of the most common greedy algorithms in the vertex coloring literature. Then, using these initial solutions, we apply two optimization techniques: a conflict minimization approach with a weighting scheme: Conflict Weighting Local Search (CWLS), a partial coloring approach with a weighting scheme: Partial Weighting Local Search (PWLS).

In Section 2, we present some mainstream algorithmic components to solve the vertex coloring problem. Section 3 present the greedy algorithms we implemented (namely the saturation degree and orientation-based greedy algorithms). Sections 4 and 5 present the local-search and weighting schemes we used for the competition (namely the conflict minimization, and vertices non-colored minimization). Finally, we compare our approaches in Section 6.

2 Related work

Multiple resolution methods were developed these last 40 years:

A first category consists of greedy algorithms. These algorithms are used to find good quality initial solutions in a short amount of time. We present two of most considered greedy approaches in Section 3.

A second category of algorithms consists of exact methods using branch-and-bound algorithms. These algorithms are complete search methods, thus are able to find an optimal solution and prove that they are indeed optimal. Such methods extend the DSATUR heuristic by allowing it to backtrack [13, 5]. Another category of exact methods (branch-and-cut-and-price) decompose the vertex coloring problem into an iterative resolution of two sub-problems [12, 8, 6]. The higher level that maintains a set of valid colors (defined by an independent set). It aims to cover all the vertices with a minimum-size set of colors, thus solving a set-covering problem (often described as the “master problem”). A lower level (solving a sub-problem) aiming to find a new valid coloring that would be promising, thus solving a maximum weight independent set problem (often described as the “pricing problem”). Such methods are usually able to find the optimal coloring for graphs with a few hundred vertices. However, the CG:SHOP 2022 competition instances involve at least a few thousands vertices, thus they appear to be not suited for this competition.

Finally, a third category of algorithms consists of local search algorithms. These methods start by an initial solution (often found by some greedy algorithm), remove some color, thus making the solution infeasible but with fewer colors, then apply some perturbations, aiming to make it feasible again. We present this approach in more details in Sections 4 and 5.

3 Finding Initial Solutions

Applying some local search scheme requires some initial solutions. We present in this section two greedy algorithms that enable us to find promising initial solutions quickly.

We use two approaches to find initial solutions. The best one is used for our other algorithms.

1. **DSATUR.** DSATUR is one of the most common greedy algorithms in the vertex coloring literature. It was introduced in 1979 [1]. It selects the vertex that has the most colors in its neighborhood, and assigns it to the first non-conflicting color until no vertex is left uncolored. Ties are broken by selecting the vertex with maximum degree. This algorithm does not take into account the specificities of the CG:SHOP 2022 competition.
2. **Segment Oriented Greedy.** We introduce a greedy algorithm that exploits the segment orientations. This algorithm is inspired from the *Recursive Largest First* algorithm, introduced in 1979 [10]. Recursive Largest first colors the graph one color at a time. The algorithm searches for an independent set of maximum size, assigns these vertices the same color, removes the newly colored vertices, and repeats until all vertices are assigned a color. The algorithm we propose (Segment Oriented Greedy) takes advantage of the fact that most of the challenge instances consist of long segments. Thus, if segments are almost parallel, it is likely that they do not intersect (thus forming an independent set). This greedy algorithm first sorts the segments by orientation, ranging from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. For each segment in this order, we try to color it using the first available color. If we fail, we introduce a new color. This algorithm is efficient, produces interesting initial solutions and takes into account the specificities of the competition.

4 Conflict Weighting Local Search (CWLS)

One classical approach for the vertex coloring involves allowing solutions with conflicting vertices (two adjacent vertices with the same color). It was introduced in 1987 [9] and called TABUCOL (more details in Section 4). It starts with an initial solution, removes a color (usually the one with the least number of vertices), and assigns uncolored vertices with a new color among the remaining ones. This is likely to lead to some conflicts (*i.e.* two adjacent vertices sharing a same color). The local search scheme selects a conflicting vertex, and tries to swap its color, choosing the new coloring that minimizes the number of conflicts. If it reaches a state with no conflict, we obtain a new best-known solution. The process is repeated until the stopping criterion is met. Originally, TABUCOL also contains some tabu mechanism to avoid cycling through some states. In this article, we refer to this local-search neighborhood as conflict-minimization. TABUCOL and its conflict-minimization neighborhood, obtained at the time excellent performance. It was later embedded in a large variety of algorithms, including many state-of-the-art algorithms (to the best of our knowledge). For instance MACOL [11] that uses TABUCOL as a mutation operator, and the GPX crossover to combine solutions.

Learning scheme. While the original TABUCOL algorithm includes this “tabu-list” mechanism to avoid cycling, it is not always sufficient, and requires some hyperparameter tuning in order to obtain a good performance on a large variety of instances. To overcome this issue, we developed a weighting scheme inspired from recent work in graph problems (namely Row Weighting Local Search for the set covering problem [7]). The idea is to penalize each conflict the algorithm introduces, so frequently occurring conflicts have a larger penalty, thus less likely to be selected in the future. This allows the algorithm to naturally diversify the solutions it obtains. Similarly to the Row Weighting Local Search algorithm, we use a simple

weighting scheme that initializes all the weights to 1 (thus one weight per edge). Each time a conflict is introduced, its weight is increased by 1. The algorithm minimizes the total weight instead of the total number of conflicts.

5 Partial Weighting Local Search (PWLS)

Another local search algorithm solving the vertex coloring problem was introduced in 2008 (PARTIALCOL). This algorithm proposes a new local search scheme that allows partial coloring (thus allowing uncolored vertices). The goal is to minimize the number of uncolored vertices. Similarly to TABUCOL, PARTIALCOL starts with an initial solution, removes one color (unassigning its vertices), and performs local search iterations until no vertex is left uncolored. When coloring a vertex, the adjacent conflicting vertices are uncolored. Then, the algorithm repeats the process until all vertices are colored, or the stopping criterion is met.

Learning scheme. We also introduce some weighting scheme for the PARTIALCOL local-search neighborhood. We assign a weight for each vertex, and the objective of the local search is to minimize the total weight of uncolored vertices. Each time a vertex becomes colored, its weight is increased by 1.

6 Results and Conclusions

All algorithms were executed on an Intel(R) Xeon(R) CPU E5-2687W v4 @ 3.00 GHz CPU, with 30 MB cache, and 768 GB RAM for 24 hours for each instance. To speed up the experiments, we ran 24 parallel runs on our machine. All algorithms have been implemented in the *rust* programming language and are publically available on a GitHub repository².

In order to compare the efficiency of our algorithms, we use the ARPD (Average Relative Percentage Deviation) as a comparison metric, which is defined as follows:

$$ARPD_{Ia} = \sum_{i \in I} \frac{M_{ai} - M_i^*}{M_i^*} \cdot \frac{100}{|I|}$$

where I is a set of instances with similar characteristics (in the same instance class), M_{ai} corresponds to the objective obtained by algorithm a on instance i . And M_i^* the reference solution objective for the instance i . The reference value is the best solution found by our team during the competition. The ARPD describes the performance of a given algorithm on a given instance class. This aggregation allows us to observe the overall behavior of some algorithm on some instance class. An ARPD close to 0 indicates that the algorithm is close to the best solutions we found during the competition (thus, the lower, the better).

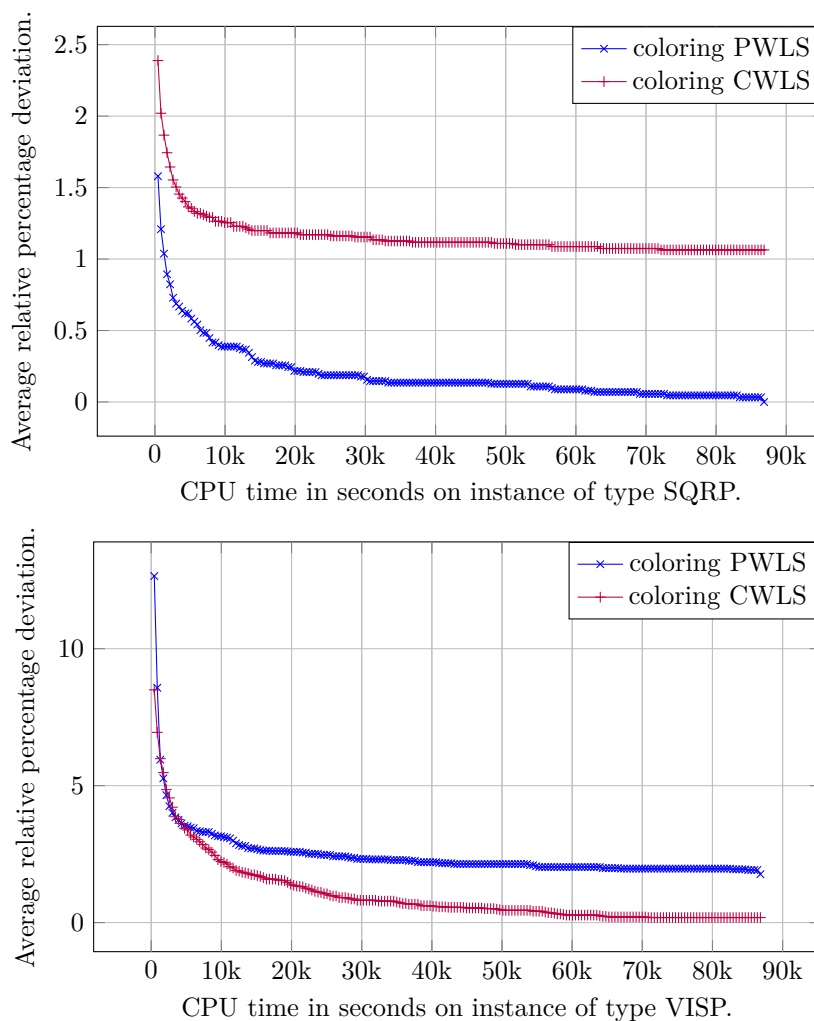
Table 1 compares all the algorithms we implemented on each class of instances. Greedy algorithms report large ARPDs, thus being far from the best-known solutions. The orientation greedy generally outperforms the DSATUR algorithm as it handles instances with long segments, where the orientation is a good conflict predictor (thus all instances but the VISP-like). DSATUR outperforms the orientation greedy on the visp-like instances (where the segment localization is more important than the orientation). Local search weighting schemes allow getting closer to the best-known solutions. It is interesting to note that CWLS and PWLS are complementary, as CWLS seems to be more efficient than PWLS on RVISP/VISP classes of instances, and PWLS being more efficient on the others. Both local search methods are able to reach almost optimal solutions (less than 2% deviation to the best-known solution in average).

² <https://github.com/librallu/dogs-color>

■ **Table 1** Average Relative Percentage Deviation comparison for all instance classes over 24h runs. Bold values indicate the algorithm that is statistically significantly better than the other one using a *Wilcoxon signed-rank test* with $\alpha = 0.05$.

Instance class	CWLS	PWLS	DSATUR	orientation greedy
reecn	1.04	0.0	24.31	16.41
rsqrp	0.51	0.11	24.03	5.89
rsqrpecn	0.0	0.0	24.86	18.08
rvisp	0.06	0.99	25.06	26.94
rvispecn	0.0	0.14	30.9	49.93
sqrp	1.06	0.0	31.22	4.9
sqrpecn	0.88	0.03	22.74	14.11
visp	0.19	1.77	29.23	27.6
vispecn	0.49	0.08	34.75	54.66

Figure 1 compares the solution quality over time of CWLS and PWLS on the SQRP and VISP instances. Both algorithms seem to be complementary depending on the instance class.



■ **Figure 1** Comparison of the performance of CWLS and PWLS over time.

From this competition, we study that adding a weighting scheme significantly improves the algorithm performance, making the algorithm more robust. This weighting allows a simple adaptive diversification mechanism. We studied two local-search neighborhoods (conflict-minimization and partial-coloring-minimization). For both of them, we presented some weighting scheme: Conflict Weighting Local Search (CWLS), and Partial Weighting Local Search (PWLS). CWLS and PWLS are both performant, and complementary depending on the instance class.

In the future, we plan to study the integration of such weighting schemes within memetic algorithms. Indeed, memetic algorithms are known to obtain excellent performance on classical vertex coloring instances.

References

- 1 Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.
- 2 Loïc Crombez, Guilherme D. da Fonseca, Yan Gerard, and Aldo Gonzalez-Lorenzo. Shadoks approach to minimum partition into plane subgraphs. In *Symposium on Computational Geometry (SoCG)*, pages 71:1–71:6, 2022.
- 3 Loïc Crombez, Guilherme D. da Fonseca, Yan Gerard, Aldo Gonzalez-Lorenzo, Pascal Lafourcade, and Luc Libralesso. Shadoks Approach to Low-Makespan Coordinated Motion Planning. In *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189, pages 63:1–63:9, 2021. doi:10.4230/LIPICs.SoCG.2021.63.
- 4 Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Stefan Schirra. Minimum partition into plane subgraphs: The CG: SHOP Challenge 2022. *CoRR*, abs/2203.xxx, 2022. arXiv:2203.07444.
- 5 Fabio Furini, Virginie Gabrel, and Ian-Christopher Ternier. An improved dsatur-based branch-and-bound algorithm for the vertex coloring problem. *Networks*, 69(1):124–141, 2017.
- 6 Fabio Furini and Enrico Malaguti. Exact weighted vertex coloring via branch-and-price. *Discrete Optimization*, 9(2):130–136, 2012.
- 7 Chao Gao, Xin Yao, Thomas Weise, and Jinlong Li. An efficient local search heuristic with row weighting for the unicost set covering problem. *European Journal of Operational Research*, 246(3):750–761, 2015.
- 8 Stefano Gualandi and Federico Malucelli. Exact solution of graph coloring problems via constraint programming and column generation. *INFORMS Journal on Computing*, 24(1):81–100, 2012.
- 9 Alain Hertz and Dominique de Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, 1987.
- 10 Frank Thomson Leighton. A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards*, 84(6):489, 1979.
- 11 Zhipeng Lü and Jin-Kao Hao. A memetic algorithm for graph coloring. *European Journal of Operational Research*, 203(1):241–250, 2010.
- 12 Anuj Mehrotra and Michael A Trick. A column generation approach for graph coloring. *INFORMS Journal on Computing*, 8(4):344–354, 1996.
- 13 Pablo San Segundo. A new dsatur-based algorithm for exact vertex coloring. *Computers & Operations Research*, 39(7):1724–1733, 2012.
- 14 André Schidler. SAT-based local search for plane subgraph partitions. In *Symposium on Computational Geometry (SoCG)*, pages 74:1–74:6, 2022. To appear.
- 15 Jack Spalding-Jamieson, Brandon Zhang, and Da Wei Zheng. Conflict-based local search for minimum partition into plane subgraphs. In *Symposium on Computational Geometry (SoCG)*, pages 72:1–72:6, 2022.

SAT-Based Local Search for Plane Subgraph Partitions

André Schidler  

TU Wien, Austria

Abstract

The Partition into Plane Subgraphs Problem (PPS) asks to partition the edges of a geometric graph with straight line segments into as few classes as possible, such that the line segments within a class do not cross. We discuss our approach *GC-SLIM*: a local search method that views PPS as a graph coloring problem and tackles it with a new and unique combination of propositional satisfiability (SAT) and tabu search, achieving the fourth place in the 2022 CG:SHOP Challenge.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases graph coloring, plane subgraphs, SAT, logic, SLIM, local improvement, large neighborhood search

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.74

Category CG Challenge

Supplementary Material *Software (Source Code)*: <https://github.com/ASchidler/coloring>

archived at `swh:1:dir:2b7057f17495a9a12cf7de4f857037c9ab0d6654`

Dataset (Results): <https://doi.org/10.5281/zenodo.6352601>

Funding FWF (P32441, W1255) and the WWTF (ICT19-065).

1 Introduction

Expressing the Partition into Plane Subgraphs Problem (PPS) in terms of graph coloring allows us to use decades of research on this important and well-researched NP-hard problem. While recent research investigated how to color massive graphs with several million vertices [11, 12], closer analysis of the used instances show that they are large but also very sparse. In contrast, the conflict graphs of this year’s challenge are smaller in size, with up to 73000 vertices, but have an edge density of up to 62% leading to conflict graphs with over 1.5 billion edges. Many established methods for graph coloring do not perform well on such dense graphs, making local search methods very appealing.

We present our approach *GC-SLIM*, a combination of propositional satisfiability (SAT) and tabu search based on the SAT-based local improvement (SLIM) meta-heuristic [9, 13, 14, 15, 16]. While a SAT-encoding of graph coloring can compute colorings for 26 of the 225 challenge instances, *GC-SLIM* scales to the largest challenge instances, improves upon established tabu search, and placed 4th overall and 2nd among student submission [7, 8, 10, 17].

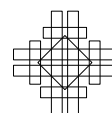
2 Preliminaries

In this paper, we only consider PPS in terms of graph coloring, i.e., we consider the conflict graph $G' = (V', E')$, containing a vertex for each line segment, and two vertices are adjacent if the corresponding line segments intersect [8]. Since *GC-SLIM* performs local search, we assume a given k -coloring $c : V(G') \rightarrow \{1, \dots, k\}$. We use the shorthands $S_\ell = \{v \in V(G') \mid c(v) = \ell\}$, $[k] = \{1, \dots, k\}$, and $N(v) = \{w \mid \{v, w\} \in E(G')\}$.



© André Schidler;
licensed under Creative Commons License CC-BY 4.0
38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 74; pp. 74:1–74:8
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



We compute the initial solution using DSATUR [6], one of the best greedy heuristics for graph coloring. DSATUR colors one vertex after another, assigning each vertex the smallest possible color that avoids monochromatic edges. DSATUR always colors the most constrained vertex next, i.e., the vertex that has the fewest viable colors available.

GC-SLIM extends the tabu search *PARTIALCOL* [5]. The idea of *PARTIALCOL* is to focus on eliminating a single color ℓ through a series of *swaps*: given a vertex v colored with ℓ and another color ℓ' , a swap changes v 's color to ℓ' and the color of all vertices in $N(v) \cap S_{\ell'}$ to ℓ . Whenever ℓ' does not occur in v 's neighborhood, the number of vertices colored with ℓ decreases. Otherwise, ℓ is propagated in the graph in the hope of success with a later swap. *PARTIALCOL* picks ℓ' among the least prevalent colors in the neighborhood of v , i.e., that minimizes $|N(v) \cap S_{\ell'}|$, with ties broken arbitrarily. The choice of ℓ' is further informed by a list of tabus for each vertex. This list contains all colors a vertex had in the last few iterations, these cannot be used for swaps to avoid getting stuck in local optima.

3 Method

We use a combination of tabu search and SAT-solving based on the SLIM method [9, 13, 14, 15, 16]. This method improves a heuristic solution through a series of local improvements, where each local improvement is accomplished by solving a smaller *local instance* with a SAT solver. Key to this method is a way to extract local instances that achieves overall improvement. For GC-SLIM the goal is eliminating color k from a given k -coloring c , thereby changing c to a $(k - 1)$ -coloring. Note that colors are interchangeable. GC-SLIM is then called again with the new coloring, until no more improvements are possible.

We use list coloring for the local instances, as it enables us to color a subgraph of G' in a way that remains compatible with the coloring outside the subgraph.

► **Definition 1** (List Coloring). *Given a k -annotated graph (G^*, L) , where $L : V(G^*) \rightarrow 2^{[k]}$, a k -list coloring is a k -coloring c for G^* , such that for all $v \in V(G^*)$ it holds that $c(v) \in L(v)$.*

Given a set $X \subseteq V(G')$ and a coloring c for G' , we create a list coloring instance (G^*, L) : $G^* = G'[X]$, the subgraph induced by X , and for each $v \in X$, we let $L(v) := [k] \setminus \{c(w) \mid w \in N(v) \setminus X\}$. Given a list coloring c' for G^* , we combine c' and c by changing $c(v)$ to $c'(v)$ for all $v \in X$. Afterwards, c is still a coloring for G' , as the lists ensure no monochromatic edges between X and $V(G^*) \setminus X$. If we are able to solve all local instances for vertices colored with k , we eventually eliminate color k from c .

We can solve the list coloring problem with a simple SAT encoding. We use for each vertex $v \in V(G^*)$ and color $\ell \in L(v)$ the variable $c_{v,\ell}$ which is true if and only if v can take color ℓ . We encode that each vertex needs a color with $\bigwedge_{v \in V(G^*)} \bigvee_{\ell \in L(v)} c_{v,\ell}$. Further, we ensure that adjacent vertices have different colors: $\bigwedge_{\{u,v\} \in E(G^*), \ell \in L(u) \cap L(v)} \neg c_{u,\ell} \vee \neg c_{v,\ell}$.

Finding good local instances is surprisingly difficult. While it was possible to find k -colorings for many local instances, eventually we would always come upon a vertex that could not be recolored, no matter how we created the local instance. We overcame this issue with inspiration taken from *PARTIALCOL*.

GC-SLIM does not try to eliminate k from the local instance. Instead, our method is satisfied with any k -coloring that is different and that minimizes the number of k -colored vertices. Given a set X of vertices that defines the local instance (G^*, L) , we perform the following two changes on L : (i) for all vertices $v \in S_k$ we remove k from $L(v)$, thereby forcing all k -colored vertices to change their color, and (ii) for all vertices $v \in X \setminus S_k$, we add k to $L(v)$. Since we eventually eliminate k , monochromatic edges using k are not an issue. Finally,

we constrain the number of vertices colored by k using a cardinality constraint [3]. Whenever we find a coloring for the local instance, we reduce the cardinality of the constraint. This follows the idea of swaps, where no vertex remains colored with k if possible, and otherwise k is propagated through the graph.

We complement this change in goal with our method for constructing local instances: starting from a single vertex $v \in S_k$, $X_0 = \emptyset$ and $X_1 = \{v\}$, GC-SLIM constructs $X = \bigcup X_i$ iteratively such that the two invariants $X \cap S_k = \{v\}$ and $|X| \leq b$ hold, i.e., v is the only vertex with color k and $|X|$ does not exceed a budget b . The set $X_{i>1}$ is constructed by adding specific neighbors of each vertex $w \in X_{i-1} \setminus X_{i-2}$, the vertices added in the last iteration. For each w , GC-SLIM adds the neighbors colored in the m least prevalent colors to X_i . The process stops when no more vertices can be added without exceeding the budget. We call m the *branching limit*, as the whole process resembles breadth-first search where the breadth in each step is limited by m . The budget ensures that the local instances stay small enough to be tackled by a SAT solver. Further, we use $\min_{\ell \in [k-1]} |c_\ell \cap N(v)|$ as the upper bound on the number of vertices colored with k in the local instance. This corresponds to a normal swap and ensures that GC-SLIM does not perform worse than PARTIALCOL.

Similar to tabu search, GC-SLIM stores the last colors of a vertex and disregards them during the construction of X , even if they are the least prevalent. We also considered removing tabu colors from the lists of the local instances. This yielded worse results as it restricts the possibilities for the SAT solver too much.

3.1 Hyperparameters

There are several hyperparameters that can severely impact GC-SLIM's performance.

The *timeout for the SAT solver* determines the time the solver has to find a solution. A large number allows for more improvements but may waste time without finding a solution. Depending on the instance and the concrete timeout, 25% to 50% of the SAT calls time out. Generally, a small timeout of 5 seconds performs well initially as it finds improvements fast, and a higher timeout can reveal more improvements once the low timeout fails.

The *iteration limit* determines the number of local instances GC-SLIM generates per color. Higher numbers increase the chance of success, but may waste time if unsuccessful. Again, low limits are good initially and higher iteration limits perform better in later stages.

The *branching limit* controls the breadth versus depth of the exploration when generating the local instance. Varying this parameter leads to different results, where smaller values of 2 and 3 find the most improvements, and values up to 15 reveal further improvements.

GC-SLIM adjusts the *budget* for the local instance automatically. Starting from an initial budget of 300 vertices, after any three consecutive SAT solver calls that time out, the budget decreases by 60 vertices. Whenever three consecutive SAT solver calls are successful, the budget increases by 60 vertices. In practice, the budget varies between 60 vertices – the lower limit – for very dense graphs and over 2000 vertices for sparse graphs.

3.2 Parallelisation

GC-SLIM uses multithreading as follows. Each thread tries to eliminate a color on its own and threads only synchronize if one of them succeeds. At this point, the successful thread shares the improved coloring and each thread starts again picking a color to eliminate.

We use multithreading in two ways: (i) to eliminate more colors in the same amount of time, and (ii) to try different hyperparameter settings, potentially on the same colors.

4 Experiments

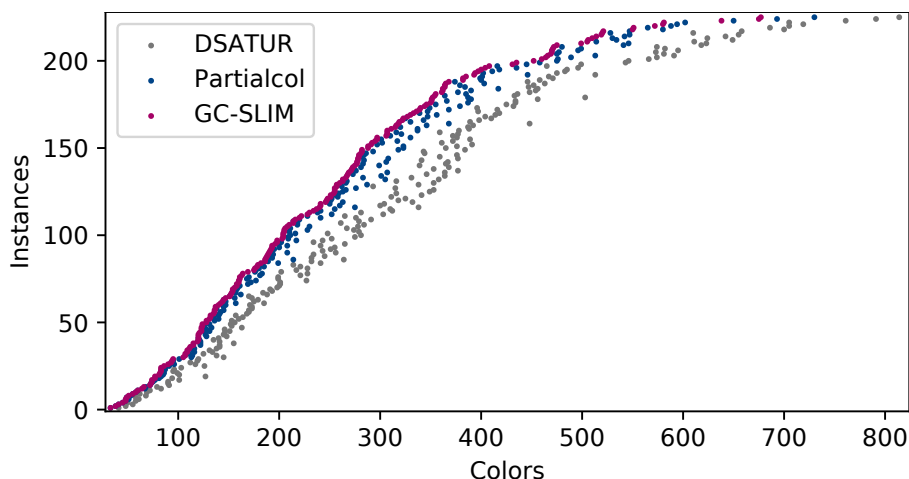
We discuss both the results from the competition and experiments with a specific time limit. GC-SLIM was implemented in C++, compiled with gcc 7.5.0, and run under Ubuntu 18.04.

The servers we used during the competition had two Xeon E5-2640v CPUs, each with 10 cores running at 2.4 GHz. The experiments ran on nodes with two AMD EPYC 7402 CPUs, each with 24 cores running at 2.8 GHz. Each run was limited to 64 GB of memory and 24 hours runtime. We used all available cores and no other experiments ran in parallel.

GC-SLIM uses the SAT solvers Glucose 3 [2] and Cadical 1.5.0 [1, 4]. Cadical is the default solver and when varying the hyperparameters, we also varied solvers.

4.1 Competition

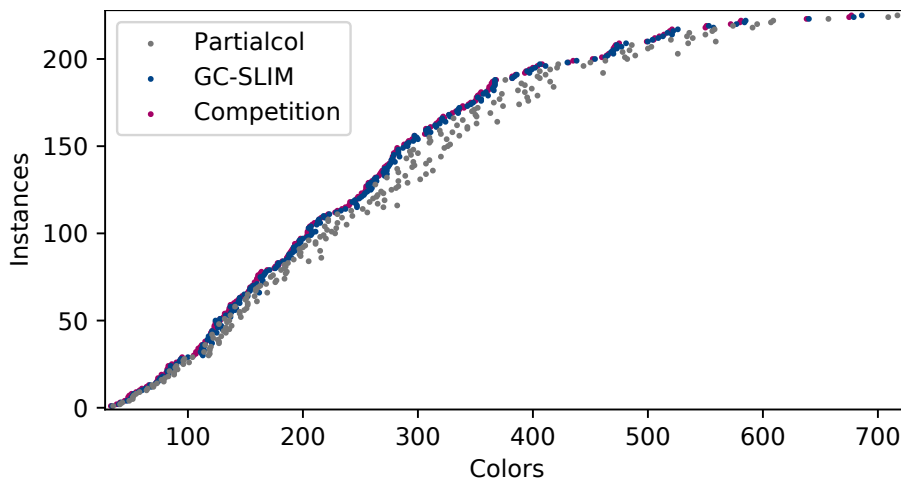
During the competition, we kept the current best coloring for each instance in a repository. This allows us to give a rough timeline of the changes, particularly on the impact of GC-SLIM. Figure 1 shows the initial number of colors, the number of colors achieved with PARTIALCOL right before introducing GC-SLIM, and the final number of colors.



■ **Figure 1** Comparison of the best colorings at different phases of the competition. The instances are ordered by the number of colors in the best result.

In the first phase, we used PARTIALCOL with varied parameters. In each iteration, the color for swaps was initially the least prevalent color, until PARTIALCOL failed to find improvements. Then, we added small random values to the color counts to diversify the exploration of the search space, until eventually, we picked random colors, which enabled further improvements. At the end of this phase, PARTIALCOL could not eliminate any color within 10 million swaps.

In the last phase, we used several runs of GC-SLIM per instance. Each run had up to twelve hours to eliminate one or more colors. For each run, we varied hyperparameters and SAT solvers. Towards the end, we used multithreading for instances with a large gap between upper bound and a clique lower bound. As Figure 1 shows, GC-SLIM was able to improve the colorings of most instances. On average, GC-SLIM removed over 50 additional colors per instance. Unfortunately, developing GC-SLIM was a long process and the final version finished too close to the deadline to achieve the best possible results.



■ **Figure 2** Comparison of 24 hour runs of tabu search and GC-SLIM. Our competition results are added as a reference. The instances are ordered by the number of colors in the best coloring.

4.2 24-hour experiment

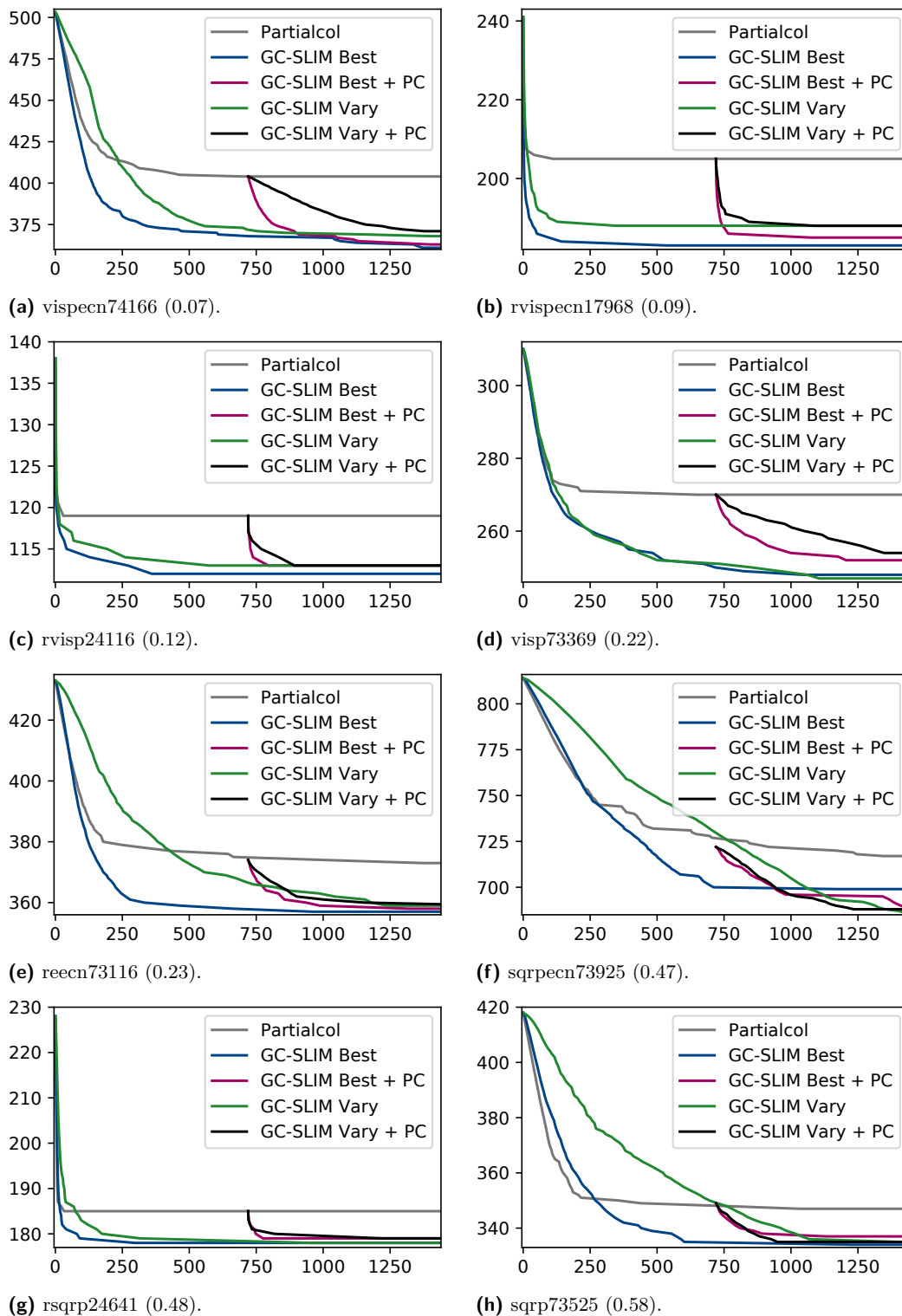
The usefulness of our method is not fully captured by the competition results, as applying local search for several weeks is usually not an option. We therefore ran PARTIALCOL and GC-SLIM with various configurations for 24 hours. For this purpose, we tried different hyperparameter settings in a shorter 5-hour run on ten instances and picked the settings that performed best as *best* configuration. The *varied* configuration uses a different set of hyperparameters in each thread. Both configurations use five parallel threads. These configurations run either alone, or combined with PARTIALCOL, where PARTIALCOL runs for twelve hours and then GC-SLIM for another twelve hours.

The overall results in Figure 2 – GC-SLIM shows the best result over all configurations – show that GC-SLIM improves upon PARTIALCOL even in this shorter timeframe. Further, the experimental results are comparable to our competition results on many instances: the competition results use only 3 fewer colors on average. This suggests the strong possibility for further improvements, given more time.

In Figure 3 we show how the number of colors develops over time for the largest instances of their respective set. We can see that the majority of improvements are achieved within a short amount of time, and PARTIALCOL quickly struggles, while GC-SLIM is able to find further improvements. Interestingly, the combination of both methods is sometimes able to find more improvements than either method alone.

5 Conclusion

GC-SLIM showed good results both in the experiments and in the competition. Key to the performance of our algorithm is the local instance selection that is closely tied to PARTIALCOL. We hope that our method will serve as a template for combining local search and SAT, and considering other local search methods will lead to further improvements.



■ **Figure 3** Color reduction (y-axis) over time (x-axis in minutes) for different configurations.

References

- 1 Cadical. <http://fmv.jku.at/cadical/>. Accessed: 2022-02-20.
- 2 Glucose. <https://www.labri.fr/perso/lsimon/glucose/>. Accessed: 2022-02-20.
- 3 Olivier Bailleux and Yacine Boufkhad. Efficient CNF encoding of boolean cardinality constraints. In Francesca Rossi, editor, *Principles and Practice of Constraint Programming - CP 2003, 9th International Conference, CP 2003, Kinsale, Ireland, September 29 - October 3, 2003, Proceedings*, volume 2833 of *Lecture Notes in Computer Science*, pages 108–122. Springer, 2003. doi:10.1007/978-3-540-45193-8_8.
- 4 Armin Biere, Katalin Fazekas, Mathias Fleury, and Maximillian Heisinger. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020. In Tomas Balyo, Nils Froleyks, Marijn Heule, Markus Iser, Matti Järvisalo, and Martin Suda, editors, *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, volume B-2020-1 of *Department of Computer Science Report Series B*, pages 51–53. University of Helsinki, 2020.
- 5 Ivo Blöchliger and Nicolas Zufferey. A graph coloring heuristic using partial solutions and a reactive tabu scheme. *Comput. Oper. Res.*, 35(3):960–975, 2008. doi:10.1016/j.cor.2006.05.014.
- 6 Daniel Brélaz. New methods to color the vertices of a graph. *Commun. ACM*, 22(4):251–256, April 1979.
- 7 Loïc Crombez, Guilherme D. da Fonseca, Yan Gerard, and Aldo Gonzalez-Lorenzo. Shadoks approach to minimum partition into plane subgraphs. In *Symposium on Computational Geometry (SoCG)*, pages 71:1–71:8, 2022.
- 8 Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Stefan Schirra. Minimum partition into plane subgraphs: The CG:SHOP Challenge 2022. *CoRR*, abs/2203.07444, 2022. arXiv:2203.07444.
- 9 Johannes K. Fichte, Neha Lodha, and Stefan Szeider. SAT-based local improvement for finding tree decompositions of small width. In *Proceedings of SAT 2017*, volume 10491 of *Lecture Notes in Computer Science*, pages 401–411. Springer Verlag, 2017.
- 10 Florian Fontan, Pascal Lafourcade, Luc Libralesso, and Benjamin Momège. Local search with weighting schemes for the CG:SHOP 2022 competition. In *Symposium on Computational Geometry (SoCG)*, pages 73:1–73:7, 2022.
- 11 Emmanuel Hebrard and George Katsirelos. A hybrid approach for exact coloring of massive graphs. In Louis-Martin Rousseau and Kostas Stergiou, editors, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research - 16th International Conference, CPAIOR 2019, Thessaloniki, Greece, June 4-7, 2019, Proceedings*, volume 11494 of *Lecture Notes in Computer Science*, pages 374–390. Springer, 2019. doi:10.1007/978-3-030-19212-9_25.
- 12 Jinkun Lin, Shaowei Cai, Chuan Luo, and Kaile Su. A reduction based method for coloring very large graphs. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 517–523. ijcai.org, 2017. doi:10.24963/ijcai.2017/73.
- 13 Neha Lodha, Sebastian Ordyniak, and Stefan Szeider. A SAT approach to branchwidth. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4894–4898. ijcai.org, 2017. doi:10.24963/ijcai.2017/689.
- 14 Vaidyanathan Peruvemba Ramaswamy and Stefan Szeider. Learning fast-inference bayesian networks. *Advances in Neural Information Processing Systems*, 34, 2021.
- 15 Vaidyanathan Peruvemba Ramaswamy and Stefan Szeider. Turbocharging treewidth-bounded Bayesian network structure learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 3895–3903. AAAI Press, 2021.

- 16 André Schidler and Stefan Szeider. SAT-based decision tree learning for large data sets. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 3904–3912. AAAI Press, 2021. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/16509>.
- 17 Jack Spalding-Jamieson, Brandon Zhang, and Da Wei Zheng. Conflict-based local search for minimum partition into plane subgraphs. In *Symposium on Computational Geometry (SoCG)*, pages 72:1–72:6, 2022.