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

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
LIPICS 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.

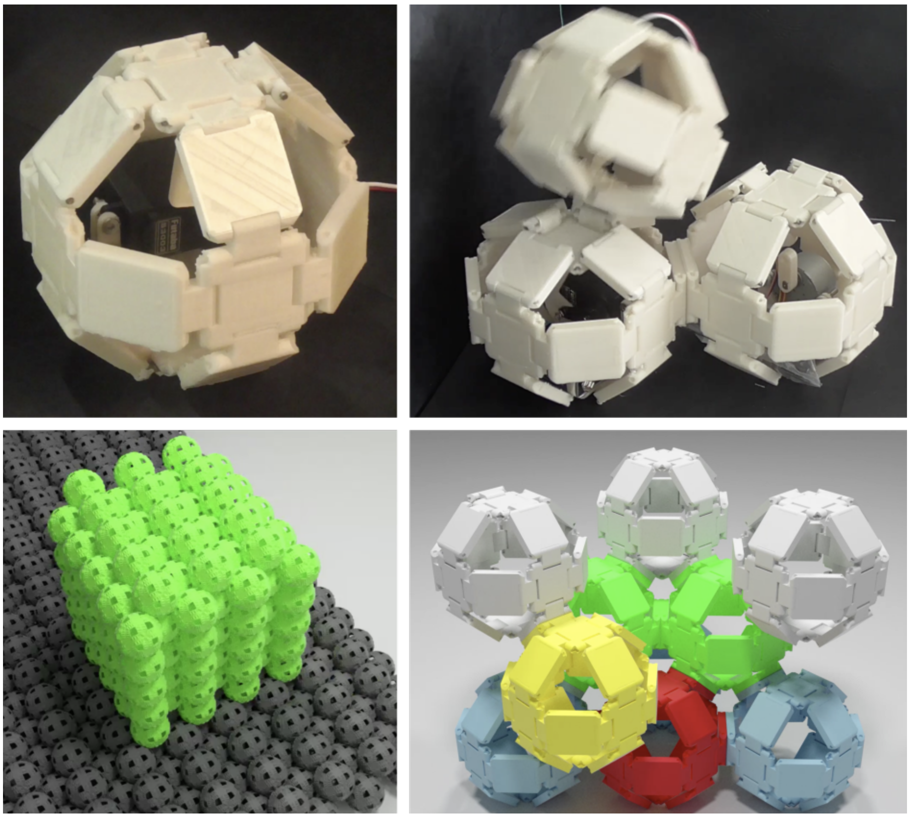

Figure 1 (Top left) A Datom. (Top right) A Datom performing a local move between neighbors. (Bottom right) A local arrangement. (Bottom left) A large-scale arrangement of 3D Catoms.

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.

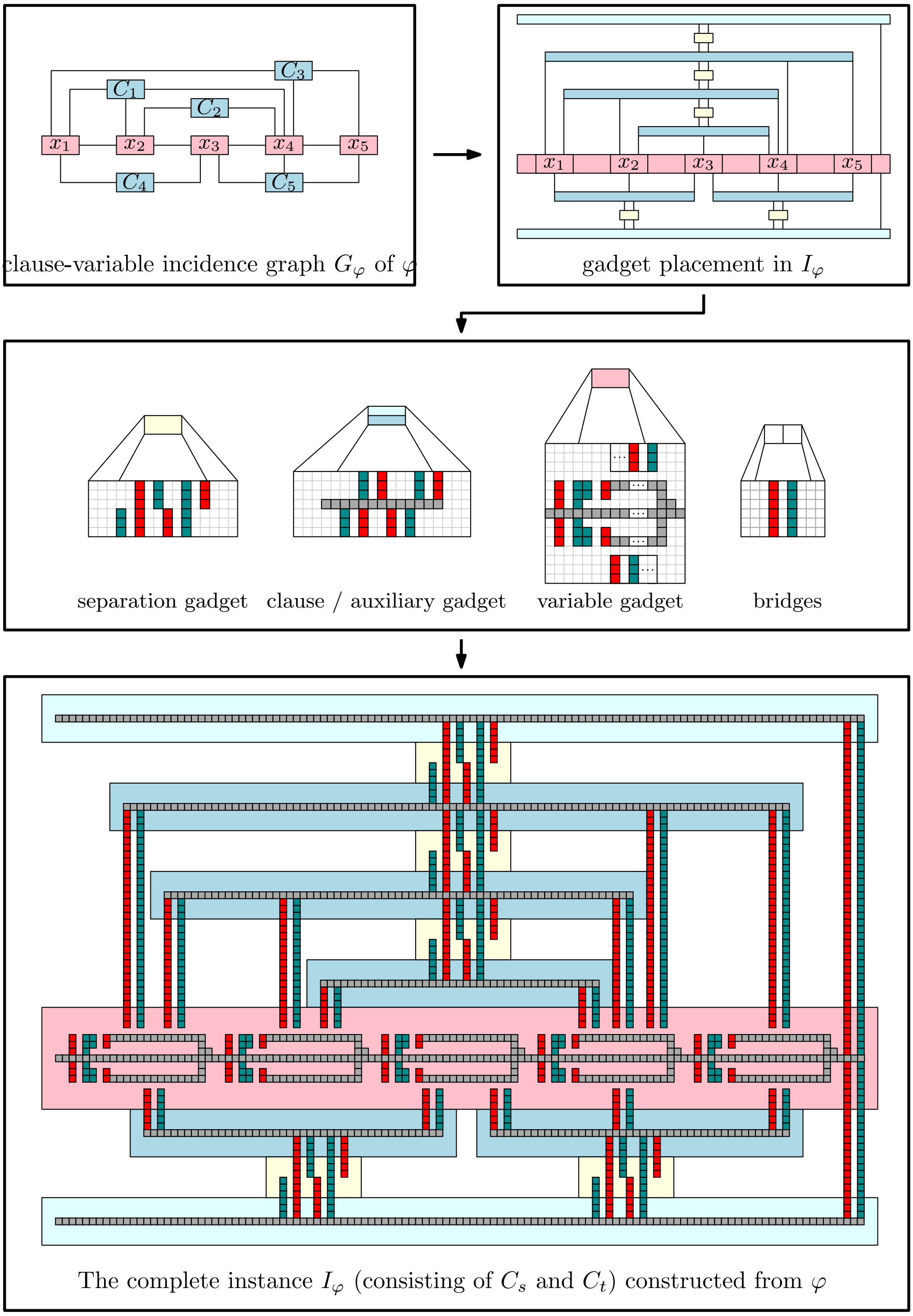

Figure 2 Symbolic overview of the NP-hardness reduction. The depicted instance is due to the PLANAR MONOTONE 3SAT formula $\varphi = (x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_4 \vee x_5) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_3} \vee \overline{x_4} \vee \overline{x_5})$. 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.

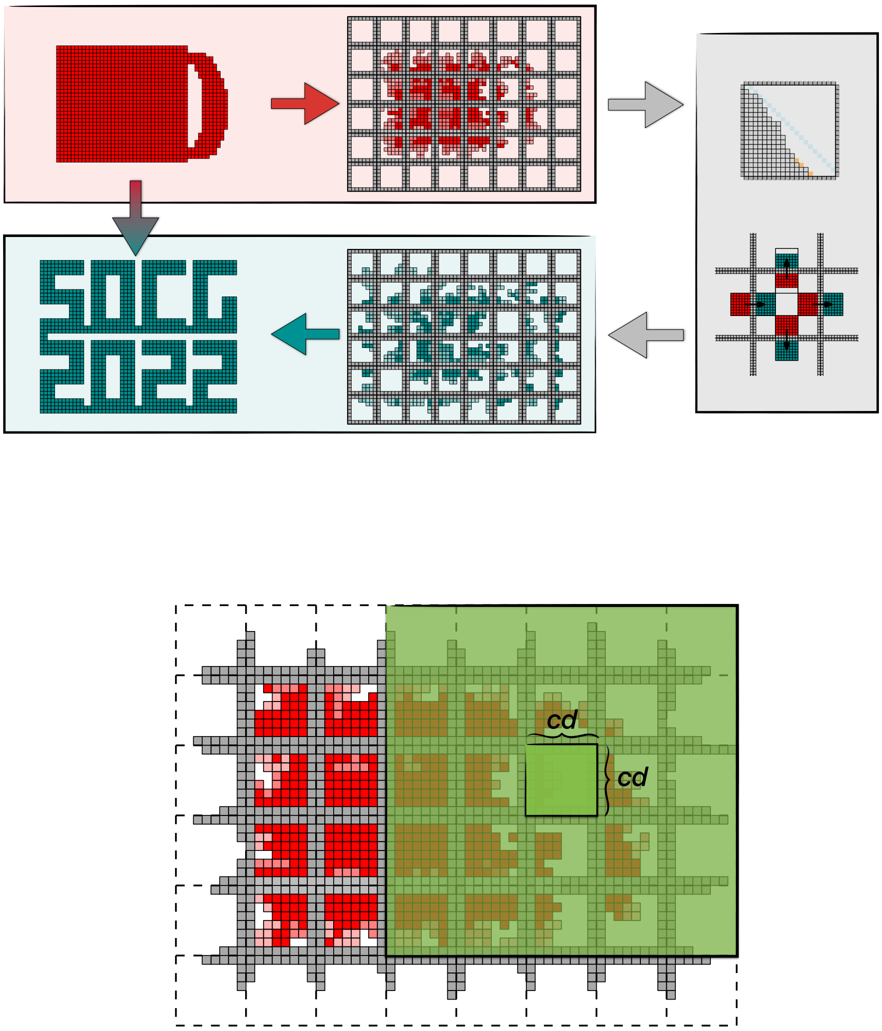

Figure 3 (Top) The algorithmic approach for achieving constant stretch while maintaining connectivity. (Bottom) Idea of the scaffold construction and tile size.

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/SFI57l5dOvk. 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.