

Algorithmic Applications of Baur-Strassen's Theorem: Shortest Cycles, Diameter and Matchings

Marek Cygan ^{*} Harold N. Gabow [†] Piotr Sankowski[‡]

Abstract

Consider a directed or an undirected graph with integral edge weights from the set $[-W, W]$, that does not contain negative weight cycles. In this paper, we introduce a general framework for solving problems on such graphs using matrix multiplication. The framework is based on the usage of Baur-Strassen's theorem and of Strojohann's determinant algorithm. It allows us to give new and simple solutions to the following problems:

Finding Shortest Cycles – We give a simple $\tilde{O}(Wn^\omega)$ time algorithm for finding shortest cycles in undirected and directed graphs. For directed graphs (and undirected graphs with non-negative weights) this matches the time bounds obtained in 2011 by Roditty and Vassilevska-Williams. On the other hand, no algorithm working in $\tilde{O}(Wn^\omega)$ time was previously known for undirected graphs with negative weights. Furthermore our algorithm for a given directed or undirected graph detects whether it contains a negative weight cycle within the same running time.

Computing Diameter and Radius – We give a simple $\tilde{O}(Wn^\omega)$ time algorithm for computing a diameter and radius of an undirected or directed graphs. To the best of our knowledge no algorithm with this running time was known for undirected graphs with negative weights.

Finding Minimum Weight Perfect Matchings – We present an $\tilde{O}(Wn^\omega)$ time algorithm for finding minimum weight perfect matchings in undirected graphs. This resolves an open problem posted by Sankowski in 2006, who presented such an algorithm but only in the case of bipartite graphs.

We believe that the presented framework can find applications for solving larger spectra of related problems. As an illustrative example we apply it to the problem of computing a set of vertices that lie on cycles of length at most t , for some t . We give a simple $\tilde{O}(Wn^\omega)$ time algorithm for this problem that improves over the $\tilde{O}(Wn^\omega t)$ time algorithm given by Yuster in 2011.

In order to solve minimum weight perfect matching problem we develop a novel combinatorial interpretation of the dual solution which sheds new light on this problem. Such a combinatorial interpretation was not known previously, and is of independent interest.

^{*}Institute of Informatics, University of Warsaw cygan@mimuw.edu.pl.

[†]Department of Computer Science, University of Colorado at Boulder hal@cs.colorado.edu.

[‡]Institute of Informatics, University of Warsaw and Department of Computer and System Science, Sapienza University of Rome sank@mimuw.edu.pl.

1 Introduction

The application of matrix multiplication to graph problems has been actively studied in recent years. The special case of unweighted graphs is well understood. For example, $\tilde{O}(n^\omega)$ time algorithms for finding shortest cycles [16] have been known for 35 years¹. But similar results for weighted graphs were obtained only last year, by Roditty and Vassilevska-Williams [25]. Their algorithm works in $\tilde{O}(Wn^\omega)$ time, where W is the largest magnitude of an edge weight. Two similar problems on weighted graphs where there has been considerable effort, but the full answer has not been achieved, are diameter and perfect weighted matching. This paper introduces a general framework that gives simple² solutions to all three of these problems and others. In the following unless otherwise stated, we work with graphs that contain edges with possibly negative weights but no negative cycles. We obtain the following results.

Finding Shortest Cycles We give a simple $\tilde{O}(Wn^\omega)$ time algorithms for finding shortest cycles in undirected and directed graphs. In the case of directed graphs the algorithm reduces the problem to one determinant computation for a polynomial matrix. On the other hand, the undirected case requires handling short 2-edge cycles in a proper way. The idea used here is an extension of the algorithm by Sankowski contained in [27], that allowed to test whether a graph contains negative weight cycle. For directed graphs (and undirected graphs with non-negative weights), our bounds match the ones obtained in 2011 by Roditty and Vassilevska-Williams [25], whereas for undirected graphs with negative weights no $\tilde{O}(Wn^\omega)$ time algorithm was previously known for this problem. For all related results see Table 1 and Table 2. For undirected graphs with negative weights the problem reduces to n computations of minimum weight perfect matchings [6].

Furthermore our algorithm for a given directed or undirected graph detects whether it contains a negative weight cycle within the same running time.

Computing Diameter and Radius We present a simple $\tilde{O}(Wn^\omega)$ time algorithm for computing a diameter and radius of undirected and directed graphs. This algorithm combines determinant computations with binary search. Since computing all pairs shortest paths suffices to find both diameter and radius $\tilde{O}(Wn^\omega)$ time algorithm follows from [31] in case of undirected graphs with non-negative weights. Moreover by generalizing the ideas of [31] used for non-negative weights by applying random sampling one can obtain the same running time for directed graphs without negative weight cycles. However, to the best of our knowledge, all previous solutions to this problem in undirected graphs with negative weights reduced the problem to n computations of minimum weight perfect matchings. For other related results see Table 3 and Table 2.

Finding Minimum Weight Perfect Matchings We present an $\tilde{O}(Wn^\omega)$ time algorithm for finding minimum weight perfect matchings in undirected graphs. This resolves an open problem posted by Sankowski in 2006 [28], who presented such an algorithm but only in the case of bipartite graphs. Some advance on this problem has been recently given by Huang and Kavitha [15], who have shown an $\tilde{O}(Wn^\omega)$ time algorithm for the maximum weight matching problem. However, the

¹The \tilde{O} notation ignores factors of $\log n$ and $\log W$. $\tilde{O}(n^\omega)$ is the time needed for a straight-line program to multiply two $n \times n$ matrices; ω is called matrix multiplication exponent.

²An objective sense in which our algorithms are simple is their use of algebra: The power of our algebraic algorithms comes from black-box routines, and the algorithms themselves use only elementary algebraic ideas.

weighted perfect matching problem is more involved and no reduction similar to the one presented in [15] is known to work. Previously, similar reduction was given in [18] for maximum weight bipartite matching. Nevertheless, to solve minimum weight perfect matching problem even in bipartite graphs one is bound to use more structured techniques. Actually, we need to develop such a technique for general graphs. We give a novel combinatorial interpretation of the dual problem. Such an interpretation for general matching problem was not known previously and is of significant independent interest. For the summary of different algorithms see Table 2.

Complexity	Author
$O(nm + n^2 \log n)$ directed	Johnson (1977) [17]
$O(n^\omega)$ nonnegative unweighted	Itai and Rodeh (1977) [16]
$O(W^{0.681} n^{2.575})$ directed	Zwick (2000) [37]
$O(nm + n^2 \log \log n)$ directed	Pettie (2004) [24]
$O(n^3 \log^3 \log n / \log^2 n)$ directed	Chan (2007) [4]
$\tilde{O}(Wn^\omega)$ directed and nonnegative undirected	Roditty and Vassilevska-Williams (2011) [25]
$\tilde{O}(Wn^\omega)$	this paper

Table 1: The complexity results for the shortest cycle problem.

Complexity	Author
$O(nm + n^2 \log n)$ directed	Johnson (1977) [17]
$O(n \min(m \log n, n^2))$ undirected	Gabow (1983) [8]
$\tilde{O}(Wn^\omega)$ nonnegative undirected	Shoshan and Zwick (1999) [31]
$O(W^{0.681} n^{2.575})$ directed	Zwick (2000) [37]
$O(n^3 \log^3 \log n / \log^2 n)$ directed	Chan (2007) [4]
$\tilde{O}(Wn^\omega)$ directed	folklore+[31]+[37]
$\tilde{O}(Wn^\omega)$	this paper

Table 3: The complexity results for the diameter and radius problem. Note that the first five results solve the more general All Pairs Shortest Paths problem.

1.1 The Framework

Our framework is based on two seminal results. First of all we use Storjohann’s algorithm [32] that computes the determinant of a degree d polynomial matrix in $\tilde{O}(dn^\omega)$ time. All the above graph problems can be encoded as a determinant problem on a polynomial matrix. However the determinant is just one number and does not provide enough information to decode the whole solution. Here we use the Baur-Strassen Theorem [2], which shows how to compute all partial derivatives of a function in the same asymptotic time as computing the function itself. For a simple constructive proof please check [22]. This theorem allows us to magnify the output of the algorithm from 1 number to n^2 numbers. The algorithms obtained in this way are very simple and work in three phases: compute the determinant of an appropriately defined matrix; apply Baur-Strassen to the result; decode the output. Even for minimum weight matching our algorithm is simple, and computes the dual solution in just a few lines of pseudocode. Based on these examples, we believe this framework will find application to a wider spectrum of problems. The full version of the paper will contain a more extensive

Complexity	Author
$O(n^2 m)$	Edmonds (1965) [5]
$O(n^3)$	Lawler (1973) [20] and Gabow (1974) [7]
$O(nm \log n)$	Galil, Micali and Gabow (1982) [13]
$O(n(m \log \log \log \frac{m}{n} n + n \log n))$	Gabow, Galil and Spencer (1984) [11]
$O(n^{\frac{3}{2}} m \log W)$	Gabow (1985) [9]
$O(n(m + n \log n))$	Gabow (1990) [10]
$O(\sqrt{n\alpha(m, n)} \log n m \log(nW))$	Gabow and Tarjan (1991) [12]
$\tilde{O}(Wn^\omega)$	this paper

Table 2: The complexity results for the minimum weight perfect matching problem. Note that first, third, fourth and fifth results solve the more general All Pairs Shortest Paths problem.

Complexity	Author
$\tilde{O}(nm)$	Suurballe and Tarjan (1984) [33]
$\tilde{O}(tWn^\omega)$ undirected	Yuster (2011) [34]
$\tilde{O}(t^{4-\omega} Wn^\omega)$ directed	Yuster (2011) [34]
$\tilde{O}(Wn^\omega)$	this paper

Table 4: The complexity results for the problem of finding vertices that lie on cycles of length at most t .

review of possible applications. Here we give one more illustration: a simple $\tilde{O}(Wn^\omega)$ time algorithm that finds every vertex that lies on a cycle of length $\leq t$, for a given arbitrary t (for a directed or undirected weighted graph, with negative edges allowed but no negative cycles). This improves the algorithms proposed by Yuster in 2011 [34] (see Table 4; those algorithms do not allow negative edges).

The paper is organized as follows. In the next two sections we introduce needed tools and give the main definitions. In Section 4 we introduce our framework and give algorithms for the shortest cycle problem in directed graphs. This motivates the introduction of the framework in Section 5. Section 6 contains the algorithm for minimum weight perfect matching problem. The next section presents the application of our framework for computing the diameter of a graph. Section 8 introduces the ideas needed to compute shortest cycles in undirected graphs with positive weights. In Section 9 we join the ideas from all previous section to solve the shortest cycle problem and the diameter problem in undirected graphs with negative weights. Finally, in Section 10 we apply our framework to find the set of vertices that lie on a cycle of length at most t .

2 Preliminaries

Our approach is based on three main ingredients.

Linear Algebra Algorithms Storjohann [32] has made an important addition to the set of problems solvable in $O(n^\omega)$ arithmetic operations: the determinant and the rational system solution for polynomial matrices.

Theorem 1 (Storjohann '03). *Let K be an arbitrary field, $A \in K[y]^{n \times n}$ a polynomial matrix of degree d , and $b \in K[y]^{n \times 1}$ a polynomial vector of the same degree. Then*

- *rational system solution $A^{-1}b$ (Algorithm 5 [32]),*
- *determinant $\det(A)$ (Algorithm 12 [32]),*

can be computed in $\tilde{O}(dn^\omega)$ operations in K , with high probability.

For the next section, note that both algorithms of Storjohann can be written as straight-line programs. The randomization does not pose a problem as it is just used at the very beginning to generate a polynomial of degree d that does not divide $\det(A)$. After this the algorithms are deterministic. Our applications will work over a finite field (see Section 2) so there is no risk of manipulating huge integers. Finally usage of the FFT to perform multiplication of degree nd polynomials does not pose a problem.

Baur-Strassen Theorem Another astonishing result is the Baur-Strassen Theorem from 1983 [2, 22]. It was used to show that matrix multiplication is no harder than determinant computation when considering algorithms that can be written as straight-line programs. Such a statement is unexpected when you realize that matrix multiplication returns n^2 numbers and determinant computation just one. However it is possible to increase the number of outputs by modifying the algorithm appropriately. Let $T(f_1, \dots, f_k)$ denote the time needed to compute functions f_1, \dots, f_k , all at the same given point.

Theorem 2 (Baur-Strassen '83). *For straight-line programs computing $f(x_1, \dots, x_n)$,*

$$T\left(f, \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}\right) \leq 5T(f).$$

Thus we can compute all partial derivatives of a function f in the same asymptotic time as is used to compute f . [22] shows a RAM routine to compute all n partials can be constructed in time $O(T(f))$ as well.

Schwartz-Zippel Lemma Our approach is to encode a graph problem in a symbolic matrix whose determinant is (symbolically) non-zero if and only if the problem has a solution. The Schwartz-Zippel Lemma [36, 30] provides the non-zero test:

Corollary 3. *For any prime p , if a (non-zero) multivariate polynomial of degree d over Z_p is evaluated at a random point, the probability of false zero is $\leq d/p$.*

We will choose primes p of size $\Theta(n^c)$ for some constant c . In a RAM machine with word size $\Theta(\log n)$, arithmetic modulo p can be realized in constant time. Moreover we would like to note that multivariate determinant expressions have been used several times in previous work, including Kirchhoff's Matrix-Tree Theorem, results of Tutte and Edmonds on perfect matchings, and recent result of Björklund [3] for undirected Hamiltonicity.

3 Definitions: Graphs with Integral Weights

A *weighted n -vertex graph* G is a tuple $G = (V, E, w, W)$, where the vertex set is given by $V = \{1, \dots, n\}$, $E \subseteq V \times V$ denotes the edge set, and the function $w : E \rightarrow [-W, W]$ ascribes weights to the edges. In this paper we consider both undirected and directed graphs. Hence, E might denote either a set of unordered pairs for undirected graphs or a set of ordered pairs for directed graphs. We define a *weight of the edge set* $F \subseteq E$ to be $w(F) = \sum_{e \in F} w(e)$.

Consider a path $p = v_1, v_2, \dots, v_k$ of length k . The *weight* of this path is simply equal to the weight of the edge set of p and denoted by $w(p)$. The *distance* from v to u in G , denoted by $dist_G(v, u)$, is equal to the minimum weight of the paths starting at v and ending in u . A path from v to u with minimum weight is called a *shortest path*. If for a path p we have $v_1 = v_k$ then the path is called a *cycle*.

In the *shortest cycle problem* we are given a weighted graph G and we need to compute the shortest (i.e., minimum weight) cycle in G . In this problem we assume that the graph contains no negative weight cycles. There is a subtle difference between the directed and undirected versions of the problem. The standard approach to reduce the undirected problem to directed problem by bidirecting the edges does not work. The resulting graph can contain cycles that pass through the same edge twice in both directions. Such cycles do not exist in the undirected graph. Moreover, when the undirected graph contains negative edges the resulting directed graph contains negative weight cycles, even if the undirected graph did not.

In the *diameter problem* we are given a weighted graph G (directed or undirected) and are asked to find the a pair of vertices $v, u \in V$ that maximizes $dist_G(v, u)$. Note that here the reduction of undirected problem to directed one by bidirecting the edges works on the in the case when the edges are positive.

A *matching* M in G is a set of edges such that each vertex is incident at most one edge from M . In a *perfect matching* each vertex is incident to exactly one matched edge. A *minimum weight perfect matching* is a perfect matching M in a weighted undirected graph that minimizes the total weight $w(M) = \sum_{e \in M} w(e)$. Many other notions of "optimum weighted matching" reduce to minimum weight perfect matching: A *maximum weight perfect matching* is equivalent to a minimum

weight perfect matching for weights $w'(e) := -w(e)$. A *minimum weight cardinality k matching* (i.e., exactly k edges are to be matched) is a minimum weight perfect matching on the graph with $n - 2k$ artificial vertices, each joined to every original vertex by a zero-weight edge. A *maximum weight matching* (i.e., we want to maximize the total weight of the matched edges) is a maximum weight perfect matching on the graph that has one artificial vertex if n is odd, plus new zero-weight edges that make the graph complete. There is a reduction going in the other direction: we set edge weights to $w'(e) := nW + w(e)$. However we are interested in time bounds that are linear in W , so this reduction is not of interest.

4 Shortest Cycles in Directed Graphs

Let K be an arbitrary field. A *symbolic polynomial* $\tilde{p}[y]$ is a multivariate polynomial over a set of variables $y \cup X$ over K . We denote the set of symbolic polynomials by $\tilde{K}[y] = K[y \cup X]$. We write as well \tilde{K} to denote the set of multivariate polynomials $K[X]$. A *symbolic matrix polynomial* $\tilde{A}[y] \in \tilde{K}[y]^{n \times n}$ is an $n \times n$ matrix whose entries are symbolic polynomials from $\tilde{K}[y]$.

We shall use a straight-line program that evaluates $\det(\tilde{A}[y])$ ($\tilde{A}[y] \in \tilde{K}[y]^{n \times n}$) using Storjohann's algorithm. Here the goal is to evaluate the determinant to a polynomial in one variable y . This program is easily constructed: Start with the original straight-line program that evaluates $\det(A)$ ($A \in K[y]^{n \times n}$) using Storjohann's algorithm. Prepend assignment statements of the form $a_{ijk} \leftarrow \tilde{a}_{ijk}$, where a_{ijk} is the variable in the original program for the coefficient of y^k in the entry A_{ij} , and the corresponding coefficient in $\tilde{A}[y]$ is $\tilde{a}_{ijk} \in \tilde{K}$. In our applications each of these new assignment statements uses $O(1)$ time, so the extra time can be ignored. Also note that all arithmetic in our straight-line programs is done in $K = Z_p$ for a chosen prime p .

We say that $\sigma : X \rightarrow K$ is an *evaluation function*. We define $\tilde{p}[y]_\sigma$ to be a one variable polynomial in y with all variables $x \in X$ substituted by $\sigma(x)$.

Let us define a *symbolic adjacency matrix* of the weighted directed graph $\vec{G} = (V, E, w, W)$ to be the symbolic matrix polynomial $\tilde{A}(\vec{G})$ such that

$$\tilde{A}(\vec{G})_{i,j} = \begin{cases} x_{i,j} y^{w((i,j))} & \text{if } (i,j) \in E, \\ 0 & \text{otherwise,} \end{cases}$$

where $x_{i,j}$ are unique variables corresponding to the edges of \vec{G} . Hence, X is the set of all variables $x_{i,j}$. For a multi-variable polynomial q , let us denote by:

- $\deg_y^*(q)$ – the degree of the smallest degree term of y in q ,
- $\text{term}_y^d(q)$ – the coefficient of y^d in q ,
- $\text{term}_y^*(q) - \text{term}_y^d(q)$ for $d = \deg_y^*(q)$.

If q is the 0 polynomial, $\deg_y^*(q) := \infty$. In the following we assume that $K := Z_p$, i.e., we work over a finite field of order p for some prime number p .

We say that a nonempty set of disjoint cycles \mathcal{C} in \vec{G} is a *cycle packing* if every vertex of \vec{G} belongs to at most one cycle in \mathcal{C} . Observe that a minimum weight cycle packing is either a shortest cycle or a collection of shortest cycles all of weight 0.

Lemma 4. *Let \vec{G} be a directed weighted graph. The minimum weight of a cycle packing in \vec{G} equals $\deg_y^*(\det(\tilde{A}(\vec{G}) + I) - 1)$. This weight is also the weight of a shortest cycle if G has no*

negative cycle. Finally for any G and any $p \geq 2$, all non-zero terms in $\det(\tilde{A}(\vec{G}) + I)$ are non-zero over a finite field Z_p .

Proof. By definition

$$\det(\tilde{A}(\vec{G}) + I) = \sum_{p \in \Gamma_n} \sigma(p) \prod_{k=1}^n (\tilde{A}(\vec{G}) + I)_{k,p_k}, \quad (1)$$

where Γ_n is the set of n -element permutations, and $\sigma(p)$ is the sign of the permutation p . A permutation p defines a set of directed edges $\mathcal{C}_p = \{(i, p_i) : 1 \leq i \leq n\}$. This edge set corresponds to a set of cycles given by the cycles of p . The edge set \mathcal{C}_p includes self-loops for all i such that $i = p_i$. Note that p corresponds to a non-zero term in the determinant if and only if \mathcal{C}_p contains only edges from E or self-loops. Hence, after throwing away self-loops p can be identified with a cycle packing in G , or \emptyset .

Let us now compute the degree of y in the term of the determinant given by p . This term is obtained by multiplying the elements of the matrix $\tilde{A}(\vec{G}) + I$ corresponding to the edges of \mathcal{C}_p . The degree of the term equals the sum of the degrees in its individual elements. These degrees encode the weights of the edges or are zero for self-loops. So the degree of the term is the total weight of the cycles in \mathcal{C}_p , where self-loops have weight zero.

The term corresponding to a set of self-loops is equal to 1. Hence, $\det(\tilde{A}(\vec{G}) + I) - 1$ contains only terms that correspond to nonempty sets of cycles. So we have proved the first assertion, i.e., the smallest degree of y is the smallest weight of a packings of cycles.

For the third assertion note that each monomial in (1) has coefficient ± 1 as each of them contains different variables. Hence each non-zero term is also non-zero over Z_p for any $p \geq 2$.

Finally for the second assertion assume \vec{G} has no negative cycle. Let C be the shortest cycle in \vec{G} . It corresponds to a cycle packing of weight $w(C)$. (The packing has a self-loop for every $v \notin C$.) In fact this is the minimum weight cycle packing. In proof take any cycle packing. If it contains more than one non-loop cycle discard all but one of them to get a packing of no greater weight (since every cycle is nonnegative). By definition this weight is $\geq w(C)$. \square

Computing the Weight of the Shortest Cycle Next we apply the ideas of the previous section to compute the weight of a shortest cycle. Since $\tilde{A}(\vec{G})$ may have entries with negative powers of y , we cannot use Storjohann's result to compute its determinant. However we can multiply it by y^W to make exponents nonnegative and use this identity:

$$\deg_y^*(\det(\tilde{A}(\vec{G}) + I) - 1) = \deg_y^*(\det((\tilde{A}(\vec{G}) + I)y^W) - y^{nW}) - nW.$$

Combining this idea with the Schwartz-Zippel lemma we obtain the following algorithm.

Algorithm 1 Computes the weight of the shortest cycle in a directed graph \vec{G} .

- 1: Generate a random substitution $\sigma : X \rightarrow Z_p$ for a prime p of order $\Theta(n^2)$.
 - 2: Compute $\delta = \det((\tilde{A}(\vec{G})|_\sigma + I)y^W) - y^{nW}$ using Storjohann's theorem.
 - 3: Return $\deg_y^*(\delta) - nW$.
-

This algorithm implies.

Theorem 5. Let $\vec{G} = (V, E, w, W)$ be a weighted directed graph without negative weight cycles. The weight of the shortest cycle in \vec{G} can be computed in $\tilde{O}(Wn^\omega)$ time, with high probability.

Note the algorithm also detects the existence of a negative cycle— it corresponds to $\delta < 0$.

Finding a Shortest Cycle The above algorithms does not seem to give the cycle by itself. There is, however, a straightforward way of doing it using the Baur-Strassen theorem. We say that edge e is *allowed* if and only if it belongs to some shortest cycle C in \vec{G} .

Lemma 6. The edge $(u, v) \in E$ is allowed if and only if: $\frac{\partial}{\partial x_{u,v}} \text{term}_y^*(\det(\tilde{A}(\vec{G})+I)-1)$ is non-zero. Moreover, for $p \geq 2$, it is non-zero over a finite field Z_p .

Proof. The above follows directly by the proof of Lemma 4. The partial derivative is non-zero if and only if the variable $x_{u,v}$ exists in some smallest degree term, and so the corresponding edge (u, v) has to lie on some shortest cycle. □

Rewrite the expression of the lemma to eliminate negative powers of y :

$$\frac{\partial}{\partial x_{u,v}} \text{term}_y^*(\det(\tilde{A}(\vec{G}) + I) - 1) = \frac{\partial}{\partial x_{u,v}} \text{term}_y^*(\det((\tilde{A}(\vec{G}) + I)y^W) - y^{nW}).$$

This implies the following algorithm and theorem.

Algorithm 2 Computes a shortest cycle in a directed graph \vec{G} .

- 1: Let $\partial_x f$ be the routine given by the Baur-Strassen Theorem to compute the matrix of partial derivatives $\frac{\partial}{\partial x_{u,v}} \text{term}_y^d \left[\det((\tilde{A}(\vec{G}) + I)y^W) - y^{nW} \right]$, where d is the degree $\deg_y^*(\delta)$ computed in Algorithm 1.
 - 2: Generate a random substitution $\sigma : X \rightarrow Z_p$ for a prime p of order $\Theta(n^2)$.
 - 3: Compute the matrix $\delta = \partial_x f|_\sigma$.
 - 4: Take any edge (u, v) such that $\delta_{u,v} \neq 0$.
 - 5: Compute the shortest path $p_{v,u}$ from v to u in \vec{G} using [27, 35].
 - 6: Return the cycle formed by (u, v) and $p_{v,u}$.
-

Note that in Step 1 we are applying the Baur-Strassen Theorem to the straight-line program for $\text{term}_y^d \left[\det((\tilde{A}(\vec{G}) + I)y^W) - y^{nW} \right]$. The latter is constructed using Storjohann's Theorem and the modification described at the start of this section.

The shortest path computation using algorithms from [27, 35] takes $\tilde{O}(Wn^\omega)$ time and succeeds with high probability, so we obtain:

Theorem 7. Let $\vec{G} = (V, E, w, W)$ be a weighted directed graph without negative weight cycles. The shortest cycle in \vec{G} can be found in $\tilde{O}(Wn^\omega)$ time, with high probability.

5 Performing Symbolic Computations

This section gives a formal description of the algebraic operations used in our algorithms. Let $\tilde{A}[y]$ be an $n \times n$ symbolic matrix and let $\sigma : X \rightarrow Z_p$ be an arbitrary evaluation function. Moreover for a prime p let $f : \tilde{Z}_p[y]^{n \times n} \rightarrow \tilde{Z}_p$ be a symbolic matrix function.

Theorem 8. *If the result of f has degree bounded by some polynomial $d_f(n)$ and $f(\tilde{A}[y]|_\sigma)$ is computable in $t_f(n)$ time, then there exists an algorithm that in $O(t_f(n))$ time checks whether $f(\tilde{A}[y])$ is symbolically non-zero over Z_p with error probability $\leq d_f(n)/p$.*

Proof. The algorithm works as follows:

1. uniformly at random choose a substitution function $\sigma : X \rightarrow Z_p$,
2. compute $f(\tilde{A}[y]|_\sigma)$ in $t_f(n)$ time,
3. check whether or not this value is zero and return the result.

For the time bound note that $t_f(n)$ is obviously $\geq |X|$. For the error bound if the degree of $f(\tilde{A}[y])$ is d , Corollary 3 shows the probability of a false zero is $\leq d/p \leq d_f(n)/p$. \square

Theorem 9. *If the result of f has degree bounded by some polynomial $d_f(n)$ and $f(\tilde{A}[y]|_\sigma)$ is computable by a straight-line program in $t_f(n) \geq d_f(n)$ time, then there exists an algorithm that in $O(t_f(n))$ time checks for each $x \in X$ whether or not $\frac{\partial}{\partial x} f(\tilde{A}[y])$ is symbolically non-zero over Z_p . All $|X|$ returned results are correct with total error probability $\leq d_f(n)|X|/p$.*

Proof. Assume that the Baur-Strassen theorem gives a routine to compute $\partial_x f := \frac{\partial}{\partial x} f(\tilde{A}[y])$ for all $x \in X$ in $O(t_f(n))$ time. The algorithm works as follows:

1. uniformly at random choose a substitution function $\sigma : X \rightarrow Z_p$,
2. apply the Baur-Strassen routine to compute $\partial_x f|_\sigma$ for all $x \in X$ in $O(t_f(n))$ time,
3. for each $x \in X$ check whether or not $\partial_x f|_\sigma$ is zero and return the result.

Obviously $\partial_x f$ has degree $\leq d_f(n)$. So Corollary 3 shows that for each $x \in X$ the probability of a false zero for $\partial_x f$ is $\leq d_f(n)/p$. Hence the union bound shows the probability of any false zero in $|X|$ results is $\leq d_f(n)|X|/p$. \square

The true goal is to check if a functional value is symbolically non-zero with high probability. This is easy to do with some weak assumptions on f . Specifically assume that any value $f(\alpha)$ in the range of f is a polynomial whose constant coefficients all have absolute value at most C , for some constant C . Also assume $t_f(n) \geq n + d_f(n)$.

Now consider the setting of Theorem 8. $f(\alpha)$ is a non-zero polynomial iff it is a non-zero polynomial over Z_p for any prime $p > C$. In Theorem 8 take p as a prime of size $\Theta(n \cdot d_f(n))$. We get error probability $O(1/n)$. The time to find p is easily accounted for by the assumption on $t_f(n)$. So for f as above, the theorem shows that in $O(t_f(n))$ time we can check if $f(\tilde{A}[y])$ is symbolically non-zero with high probability.

Next consider the setting of Theorem 9. Any value of $\partial_x f$ is a non-zero polynomial iff it is a non-zero polynomial over Z_p for any prime $p > Cd_f(n)$. In Theorem 9 take p as a prime of size $\Theta(n|X| \cdot d_f(n))$. We get error probability $O(1/n)$. The time is similar. So for f as above the theorem shows that in $O(t_f(n))$ time we can check if $\frac{\partial}{\partial x} f(\tilde{A}[y])$ is symbolically non-zero with high probability.

6 Minimum Weight Perfect Matching

This section presents an algorithm that, given an undirected graph with integral edge weights in $[0, W]$, finds a minimum weight perfect matching in $\tilde{O}(Wn^\omega)$ time, assuming such matchings exist.

The algorithm works in three phases:

1. The first phase uses algebra to reduce the problem to connected graphs, where each edge belongs to some minimum weight perfect matching (Algorithm 3). Moreover for each vertex v , we are given the value $w(M(v))$ – the minimum weight of a matching with exactly 2 unmatched vertices, one of which is v (Algorithm 4). This phase uses $\tilde{O}(Wn^\omega)$ time and succeeds with high probability.
2. The second phase defines a new weight $w'(uv) := w(uv) + w(M(u)) + w(M(v))$ for each edge uv . It performs a simple graph search algorithm on these new edges to obtain a laminar family of blossoms, which is the support of some optimum dual solution (Algorithm 5). Each blossom induces a factor critical graph. This phase is deterministic and uses $\tilde{O}(n^2)$ time.
3. The last phase uses a maximum cardinality matching algorithm (for unweighted graphs), guided by the structure of the blossoms, to obtain a minimum weight perfect matching (Lemma 20). This phase uses $\tilde{O}(n^\omega)$ time and succeeds with high probability.

To elaborate on the second phase (which in our opinion is the most interesting), let A be the set of distinct values of the weight function w' . Section 6.2 proves $|A| = O(n)$. For $\alpha \in A$ we define a 'threshold graph' $G_\alpha = (V, E')$, which is an unweighted undirected graph with $E' = \{uv \in E : w'(uv) \leq \alpha\}$. The nontrivial connected components of all the graphs G_α constitute the blossoms of an optimum dual solution! (A connected component is *nontrivial* if it has more than 1 and less than n vertices.) Our proof of this result hinges on showing there exists a special dual solution (called balanced critical dual solution) in which it is easy to find the blossoms (Lemma 28).

Figure 1 depicts a sample graph and illustrates the steps for obtaining the laminar family of blossoms.

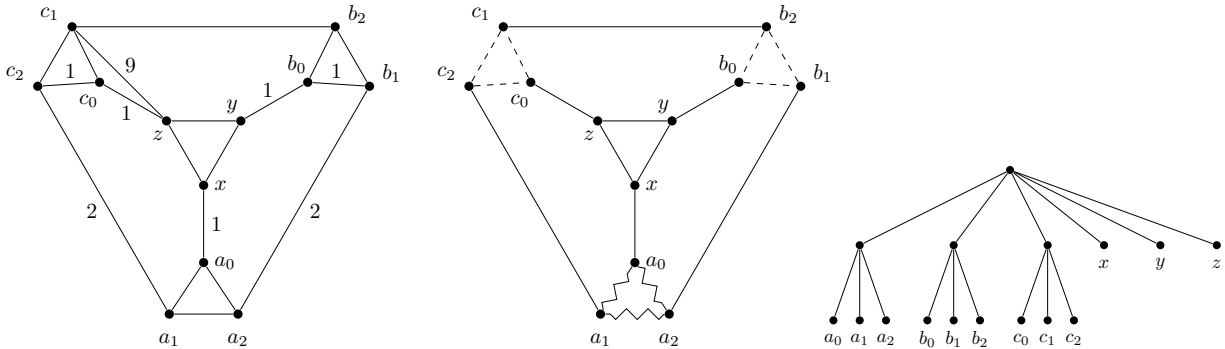


Figure 1: The matching algorithm: The far left shows an example graph. Edges without a label weigh 0. The minimum weight of a perfect matching is 3. $w(M(v))$ is 2 for $v \in \{b_2, c_1, x, y, z\}$ and 1 otherwise. The middle figure shows the allowed edges (edge c_1z was removed) and new edge weights $w'(uv) = w(uv) + w(M(u)) + w(M(v))$; edges with $w'(e) = 2$ are drawn zigzag, $w'(e) = 3$ are dashed, and $w'(e) = 4$ are straight. The far right shows the laminar family induced by the blossoms found using the threshold graphs of Algorithm 5.

In the rest of this section we show how to obtain the set of *allowed edges*, i.e., edges belonging to at least one minimum weight perfect matching and values $w(M(u))$ (Section 6.1). Next, in Section 6.2 we present the standard LP formulation of the problem. We recall and extend properties of a dual solution, in order to prove the correctness of our simple method of obtaining blossoms of an optimum dual solution. Finally, in Section 6.3, we gather all the theorems and formally prove correctness and bound the running time of our algorithm.

We would like to note, that the simplicity of our algorithm for contracting the set of blossoms from the values $w(M(u))$ is due to the fact that the hardness is hidden in the proof of the purely combinatorial existential lemmas from Section 6.2 and in the algorithm for finding unweighted maximum matching problem.

6.1 Algebraic Tools

Let us define a *symbolic adjacency matrix* of the weighted undirected graph $G = (V, E, w, W)$ to be the $n \times n$ matrix $\tilde{A}(G)$ such that

$$\tilde{A}(G)_{i,j} = \begin{cases} x_{i,j}y^{w(ij)} & \text{if } ij \in E \text{ and } i < j, \\ -x_{j,i}y^{w(ij)} & \text{if } ij \in E \text{ and } i > j, \\ 0 & \text{otherwise,} \end{cases}$$

where $x_{i,j}$ are unique variables corresponding to the edges $ij \in E$ of G . Karp, Upfal and Wigderson [19] proved that the smallest degree of y in $\det(\tilde{A}(G))$ is twice the weight of a minimum weight perfect matching in G . By using this line of reasoning together with results of Storjohann and Baur-Strassen, we show how to obtain the set of edges which appear in at least one minimum weight perfect matching.

Lemma 10. *An edge $ij \in E$ belongs to some minimum weight perfect matching iff*

$$\frac{\partial}{\partial x_{i,j}} \text{term}_y^* [\det(\tilde{A}(G))] \neq 0.$$

Proof. By the definition of a determinant we have:

$$\det(\tilde{A}(G)) = \sum_{p \in \Gamma_n} \text{sgn}(p) \prod_{k=1}^n \tilde{A}(G)_{k,p_k}, \quad (2)$$

where Γ_n is the set of n -element permutations. A permutation p defines a multiset of edges $\mathcal{C}_p = \{\{i, p_i\} : 1 \leq i \leq n \text{ and } i \neq p_i\}$. This edge multiset corresponds to a cycle cover given by the cycles of p . Reversing an odd cycle in a permutation does not change its sign, but it changes the sign of the monomial corresponding to this permutation in the determinant. Consequently, the polynomial $\det(\tilde{A}(G))$ contains only monomials corresponding to even-cycle covers of G . (An *even-cycle cover* has no odd cycles.) Since each even-cycle cover is easily decomposable into two perfect matchings, and doubling a matching gives an even-cycle cover, we infer that an edge $ij \in E$ belongs to some minimum weight perfect matching in G iff it appears in some minimum degree monomial in $\det(\tilde{A}(G))$. The lemma follows. \square

Note that an even-cycle cover corresponding to a matching – i.e., every cycle has length 2 – corresponds to a unique permutation in (2). Thus the lemma is true in any field Z_p .

The combination of Lemma 10 with Theorem 9 proves the following corollary.

Corollary 11. *For a weighted undirected graph $G = (V, E, w, W)$ one can compute the set of edges which belong to at least one minimum weight perfect matching in $\tilde{O}(Wn^\omega)$ running time, with high probability.*

Algorithm 3 Computes the set of allowed edges in the graph G .

- 1: Generate a random substitution $\sigma : X \rightarrow Z_p$ for a prime p of order $\Theta(n^2)$.
 - 2: Compute $d = \deg_y^* \left[\det(\tilde{A}(G)|_\sigma) \right]$ using Storjohann's theorem.
 - 3: Let $\partial_x f$ be the routine given by the Baur-Strassen theorem to compute the matrix of partial derivatives $\frac{\partial}{\partial x_{i,j}} \text{term}_y^d \left[\det(\tilde{A}(G)) \right]$.
 - 4: Generate a random substitution $\sigma : X \rightarrow Z_p$ for a prime p of order $\Theta(n^4)$.
 - 5: Compute the matrix $\delta = \partial_x f|_\sigma$.
 - 6: Mark each edge ij , where $i < j$, as allowed if $\delta_{i,j} \neq 0$.
-

Definition 12 ($\mathbf{M}(\mathbf{uv})$, $\mathbf{M}(\mathbf{u})$). *For a pair of vertices $u, v \in V$ let $M(uv)$ be a minimum weight perfect matching in $G \setminus \{u, v\}$, i.e., G with vertices u and v removed. Similarly, for a vertex u let $M(u)$ be a minimum weight almost-perfect matching in the graph $G \setminus \{u\}$.*

Note that $M(u)$ always exists, since we assume the given graph G has a perfect matching. In contrast $M(uv)$ needn't exist. In that case $M(uv)$ is ∞ .

Lemma 13. *Let $G = (V, E, w, W)$ be a weighted undirected graph. Then $\deg_y^*(\text{adj}(\tilde{A}(G))_{i,j}) = w(M) + w(M(ij))$.*

Proof. We have $\text{adj}(\tilde{A}(G))_{i,j} = (-1)^{i+j} \det(\tilde{A}(G)^{j,i})$. Equivalently, if we take \tilde{Z} to be the matrix obtained from $\tilde{A}(G)$ by zeroing entries of the j 'th row and the i 'th column and setting the entry (j, i) to 1, then $\text{adj}(\tilde{A}(G))_{i,j} = \det(\tilde{Z})$, and so

$$\text{adj}(\tilde{A}(G))_{i,j} = \sum_{p \in \Gamma_n} \text{sgn}(p) \prod_{k=1}^n z_{k,p_k}. \quad (3)$$

The permutation p can be viewed as a set of directed cycles \mathcal{C} covering G , where there is a cycle c that contains the edge (j, i) . (By definition (j, i) is an edge of \mathcal{C} , even when $(j, i) \notin E(G)$. (j, j) is an example. The other edges of \mathcal{C} are in $E(G)$.)

We claim that the terms in this adjoint correspond to even-length-cycle covers that contain a cycle c through (j, i) . In other words cycle covers that contain an odd cycle make no net contribution to (3). In proof let d be an odd cycle in \mathcal{C} . First suppose $d \neq c$. Reversing its direction changes the sign of its contribution to (3), from antisymmetry of $\tilde{A}(G)$. So such covers do not contribute to the adjoint. Second suppose $d = c$. G has an even number of vertices. Thus \mathcal{C} has an even number of edges, and c cannot be the unique odd cycle.

Now take any \mathcal{C} contributing to (3). \mathcal{C} decomposes into two matchings by taking alternate edges from each (even) cycle. One matching, say N , is a perfect matching of G ; the other is a perfect matching of $G \setminus \{i, j\}$, say $N(ij)$, plus edge (j, i) . We conclude that the adjoint of (3) is 0 if N or $N(ij)$ does not exist. This proves the case of the lemma when M or $M(ij)$ does not exist.

Now assume both M and $M(ij)$ exist. The degree of any term in (3) equals the sum of the edge weights in \mathcal{C} (note that the weight of the edge (j, i) is considered as 0, since $z_{j,i} = 1$). Take

any term that has the smallest degree in (3). Its degree is

$$\deg_y^*(\text{adj}(\tilde{A}(G))_{i,j}) = w(N) + w(N(ij)) \geq w(M) + w(M(ij)).$$

We will complete the proof by showing

$$\deg_y^*(\text{adj}(\tilde{A}(G))_{i,j}) \leq w(M) + w(M(ij)).$$

(Obviously the two displayed inequalities show equality holds, so they imply the lemma.) The multiset $M \cup M(ij) \cup (j, i)$ gives an even-length-cycle cover \mathcal{C}_M of G which contains the edge (j, i) . Wlog the cycle through (j, i) is the only cycle in \mathcal{C}_M (since we can make M and $M(ij)$ identical on any other even cycle). We claim the monomial corresponding to \mathcal{C}_M occurs exactly once in (3). This claim shows the monomial does not get cancelled, thus proving the desired inequality.

To prove the claim, a variable x_{gh} appearing in any term comes from the cycle cover edge (g, h) or (h, g) . This implies that two terms with the same variables are cycle covers that differ only in the orientation of some cycles. A cycle of \mathcal{C}_M either has length two or contains (j, i) . The former does not change when it is reversed, and the latter cannot be reversed. So no other cycle cover corresponds to the monomial of \mathcal{C}_M .

Note that since the monomial for \mathcal{C}_M occurs exactly once, the lemma holds for any field Z_p . \square

If M or $M(ij)$ does not exist, the expression of the lemma equals ∞ . The lemma and the following extension also hold in any field Z_p .

Corollary 14. *The vector of values $w(M) + w(M(i))$, $i \in V$ equals $\deg_y^*(\text{adj}(\tilde{A}(G))b)$ for b a vector of n indeterminates $b = (b_1, b_2, \dots, b_n)$.*

Proof. Lemma 13 shows $w(M) + w(M(i)) = \min_j \deg_y^*(\text{adj}(\tilde{A}(G))_{i,j})$. (Note that entries $\text{adj}(\tilde{A}(G))_{i,j}$ corresponding to the 0 polynomial cause no problem.) The i th component of the vector $\text{adj}(\tilde{A}(G))b$ is $\sum_j \text{adj}(\tilde{A}(G))_{i,j} b_j$, and because of the indeterminates b_j no terms cancel when the sum is formed. Thus $\deg_y^*(\text{adj}(\tilde{A}(G))b) = w(M) + w(M(i))$. \square

This leads to the following algorithm to compute $w(M(u))$ for all $u \in V(G)$. Let $B = \{b_1, \dots, b_n\}$.

Algorithm 4 Computes the values $w(M(i))$ for all vertices i in the graph G .

- 1: Generate a random substitution $\sigma : X \cup B \rightarrow Z_p$ for a prime p of order $\Theta(n^3)$.
 - 2: Compute $w(M) = \deg_y^*(\det(\tilde{A}(G)|_\sigma))/2$ and $v = \det(\tilde{A}(G)|_\sigma)(\tilde{A}^{-1}(G)b)|_\sigma$ using Storjohann's theorem, where $b = (b_1, \dots, b_n)$.
 - 3: For each $i \in V$ set $w(M(i)) = \deg_y^*(v_i) - w(M)$.
-

To see this algorithm is correct Statement 2 computes $v = (\text{adj}(\tilde{A}(G))b)|_\sigma$. So Corollary 14 shows the algorithm is correct if there are no false zeroes. A rational expression (like those in $(\tilde{A}^{-1}(G)b)|_\sigma$) is zero if and only if its numerator is zero and its denominator is nonzero. So we can apply the Schwartz-Zippel Lemma to show the final products have no false zeroes (in their lowest order term). So each $\deg_y^*(v_i)$ is computed correctly with high probability.

Regarding efficiency consider the n multiplications of degree nW polynomials done to form v in Statement 2. We only use the lowest degree term of each product (Statement 3). That term comes from the lowest degree term in $\det(\tilde{A}(G)|_\sigma)$ and the lowest degree term in the numerator and the denominator of $(\tilde{A}^{-1}(G)b)|_\sigma$. So we can find the smallest degree of y that corresponds to $\deg_y^*(v_i)$ using $O(1)$ additions and subtractions, without multiplying polynomials.

We conclude:

Corollary 15. *Algorithm 4 computes the values $w(M(u))$, for all $u \in V(G)$, in $\tilde{O}(Wn^\omega)$ time, with high probability.*

6.2 Properties of the Dual

We move on to the linear programming formulation of the minimum weight perfect matching problem given by Edmonds [5]. An *odd set* has odd cardinality; Ω denotes the collection of odd subsets of V of cardinality ≥ 3 .

$$\begin{aligned} \min \sum_{e \in E} w(e)x_e \\ x(\delta(v)) &= 1, \text{ for all } v \in V \\ x(\delta(U)) &\geq 1, \text{ for all } U \in \Omega \\ x_e &\geq 0, \text{ for } e \in E \end{aligned} \tag{4}$$

The variables x_e indicate when an edge is included in the solution. Here, $\delta(U)$ denotes all edges $uv \in E$ having $|\{u, v\} \cap U| = 1$. We write $\delta(u)$ for $\delta(\{u\})$ and $x(F)$ for $\sum_{e \in F} x_e$.

The dual problem has variables π_v for each vertex v and π_U for each odd set U :

$$\begin{aligned} \max \sum_{v \in V} \pi_v + \sum_{U \in \Omega} \pi_U \\ \pi_u + \pi_v + \sum_{U \in \Omega, uv \in \delta(U)} \pi_U \leq w(uv) \text{ for all } uv \in E \\ \pi_U \geq 0 \text{ for all } U \in \Omega \end{aligned} \tag{5}$$

We say that an edge $e = uv$ is *tight* with respect to a dual π if equality holds in (5). A *laminar family* is a set system where each pair of sets is either disjoint or one set contains the other. Moreover, a graph is *factor critical* if after removing each vertex the graph has a perfect matching. We use existence of the following dual:

Lemma 16 (Edmonds '65 [5]). *There exists an optimal dual solution $\pi : V \cup \Omega \rightarrow \mathbb{R}$, such that:*

1. *the set system $\{U \in \Omega : \pi_U > 0\}$ forms a laminar family,*
2. *for each $U \in \Omega$ with $\pi_U > 0$, the graph $G[U]$ with each set of $\{S \in \Omega : S \subset U, \pi_S > 0\}$ contracted is factor critical.*

Definition 17 (critical dual, blossom). *An optimum dual solution satisfying the conditions from Lemma 16 is a critical dual solution. A set $U \in \Omega$ such that $\pi_U > 0$ is a blossom w.r.t. π .*

Blossoms of critical dual solutions have the following useful property (note the lemma below is weight-oblivious and the only input given to the algorithm is an undirected unweighted graph, the family of blossoms, and v).

Lemma 18. *Consider any critical dual solution and let $U \in \Omega$ be an arbitrary blossom. For any vertex $v \in U$ there exists a perfect matching $M(U, v)$ in $G[U \setminus \{v\}]$, such that for each blossom $U_0 \subseteq U$, $|M(U, v) \cap \delta(U_0)|$ is 0 if $v \in U_0$ and 1 if $v \notin U_0$. Furthermore, given the family of all blossoms and v , one can find such a matching in $\tilde{O}(|U|^\omega)$ running time, with high probability.*

Proof. Let \mathcal{B} be the set of blossoms of π properly contained in U (\mathcal{B} might be empty); moreover let \mathcal{B}_{\max} be the set of inclusionwise maximal sets in \mathcal{B} . Let G' be the graph $G[U]/\mathcal{B}_{\max}$ and let v' be a vertex of G' corresponding to v . (Here we use the contraction operator – if \mathcal{S} is a family of disjoint vertex sets, G/\mathcal{S} denotes the graph G with each set of \mathcal{S} contracted to a single vertex.)

Initially let $M(U, v) \subseteq E$ be a perfect matching in $G' \setminus v'$. It exists since G' is factor critical. For each blossom $U_0 \in \mathcal{B}_{\max}$, recursively find a perfect matching M_0 in the graph $G[U_0 \setminus x]$, where x is the single vertex of the intersection of U_0 and $V(M(U, v)) \cup \{v\}$. Add the edges of M_0 to $M(U, v)$.

By construction the final matching $M(U, v)$ satisfies conditions from the lemma. For the time bound note that the laminarity of \mathcal{B} implies the total number of vertices in all graphs constructed by the above procedure is $O(n)$. The algorithms of [23, 26, 14] find a perfect matching on an arbitrary graph of n vertices in time $\tilde{O}(n^\omega)$, with high probability. Hence our recursive procedure runs in total time $\tilde{O}(|U|^\omega)$. \square

Complementary slackness gives the following observation.

Observation 19. *For any optimum dual solution:*

- (a) *a set $U \in \Omega$ with $\pi_U > 0$ has exactly one edge of $\delta(U)$ in any minimum weight perfect matching;*
- (b) *an edge belonging to any minimum weight perfect matching is tight.*

Lemma 20. *Given a weighted undirected graph $G = (V, E, w, W)$ where each edge is allowed, and the set of blossoms \mathcal{B} of some critical dual solution, one can find a minimum weight perfect matching in $\tilde{O}(n^\omega)$ time, with high probability.*

Proof. Let $\mathcal{B}_{\max} \subseteq \mathcal{B}$ be the set of inclusionwise maximal blossoms. Let graph $G' = G/\mathcal{B}_{\max}$. G' has a perfect matching M_0 (since Observation 19(a) shows any minimum weight perfect matching in G contains a subset of edges forming a perfect matching in G').

We extend M_0 to a perfect matching in G by considering the blossoms of $U \in \mathcal{B}_{\max}$ one by one. Let v_U be the unique vertex of U that is matched by M_0 . Add the matching $M(U, v_U)$ of Lemma 18 to M_0 . The final set M_0 is a perfect matching for G .

The time for this procedure is $\tilde{O}(n^\omega)$. This follows exactly as in Lemma 18, since the total number of vertices in all graphs considered is $O(n)$.

Finally, we prove that M is a minimum weight perfect matching in G by showing that for each blossom $U \in \mathcal{B}$ the set M contains exactly one edge of $\delta(U)$. Consider a maximal blossom $U \in \mathcal{B}_{\max}$. When finding a perfect matching in G' we have added exactly a single edge of $\delta(U)$ to the set M . Moreover each edge of $M(U', v_{U'})$, for any maximal blossom $U' \in \mathcal{B}_{\max}$, is contained in U' , and therefore the set M contains exactly one edge of $\delta(U)$. Now let us consider a blossom $U \in \mathcal{B} \setminus \mathcal{B}_{\max}$, and let $U' \in \mathcal{B}_{\max}$ be a maximal blossom containing U . If in the first phase (finding a perfect matching in G') we added no edge from $\delta(U)$ to the set M , then by Lemma 18 in the set $M(U', v_{U'})$ there is exactly one edge of $\delta(U)$, whereas for each other maximal blossom $U'' \in \mathcal{B}_{\max}, U'' \neq U'$, in the set $M(U'', v_{U''})$ there is no edge of $\delta(U)$. If, however, in the first phase we added an edge from $\delta(U)$ to the set M , then by the choice of $v_{U'}$, which is the endpoint of the edge of $\delta(U) \cap M$, no other edge of $\delta(U)$ is added to the set M .

Since all the edges are allowed, by Observation 19(b) the sum of values of edges of M is equal to the cost of the critical dual solution, which proves that M is a minimum weight perfect matching in the graph G . Note that our algorithm does not need exact values of the dual nor even weights of edges, since its input is only graph G and set \mathcal{B} . \square

A critical dual solution gives rise to a weighted tree in a natural way:

Definition 21 (dual tree). *Let $\pi : \Omega \cup V \rightarrow \mathbb{R}$ be a critical dual solution, with \mathcal{B} the set of its blossoms. The dual tree $T(\pi)$ is a rooted tree on nodes $\{V\} \cup \mathcal{B} \cup V$, where V is the root, vertices of V are leaves, blossoms of \mathcal{B} are internal nodes and the parent-child relation in $T(\pi)$ is naturally defined inclusionwise. The weight of the edge from a node $t \in \mathcal{B} \cup V$ to its parent is π_t . The height of the tree $H(T(\pi))$ is the weight of a longest path from the root to some leaf.*

In this definition note that the last edge of a path defining $H(T(\pi))$ may have negative length. For a tree T with weighted edges and two nodes u, v , $dist_T(u, v)$ denotes the weight of the path between u and v . The following simple lemma provides a basic tool.

Lemma 22. *If π is a critical dual solution for a weighted graph $G = (V, E, w, W)$, any allowed edge uv satisfies $w(uv) = dist_{T(\pi)}(u, v)$.*

Proof. Since uv is tight (Observation 19(b)), $w(uv) = \pi_u + \pi_v + \sum_{U \in \Omega, uv \in \delta(U)} \pi_U$. The right-hand side gives $dist_{T(\pi)}(u, v)$ for two reasons: The edges of $T(\pi)$ incident to leaves are weighted with the singleton values of π . A blossom B of π contains exactly one endpoint of the edge uv if and only if the path between u and v in $T(\pi)$ contains the edge between B and its parent. \square

The next steps of our development (Lemmas 25–27) can be derived using an appropriate version of Edmonds’ weighted matching algorithm (e.g., [29]). Here we will use a structural approach, based on the following properties of allowed edges given by Lovász and Plummer.

Lemma 23 ([21], Lemma 5.2.1 and Theorem 5.2.2). *Let $G = (V, E)$ be an undirected connected graph where each edge belongs to some perfect matching. Define a binary relation $R \subseteq V \times V$ by $(u, v) \in R$ if and only if $G \setminus \{u, v\}$ has no perfect matching. Then*

- R is an equivalence relation;
- each equivalence class of R is an independent set;
- for each equivalence class S of R , the graph $G \setminus S$ has exactly $|S|$ connected components, each of which is factor critical.

We will use a special type of critical dual solution that we call ”balanced”.

Definition 24 (balanced critical dual). *Let $\pi : \Omega \cup V \rightarrow \mathbb{R}$ be a critical dual solution, and let G' be the graph G with each blossom of π contracted. π is a balanced critical dual solution if there are two distinct vertices $u, v \in V$ such that $dist_{T(\pi)}(u, V) = dist_{T(\pi)}(v, V) = H(T(\pi))$ and further, $G' \setminus \{u', v'\}$ has a perfect matching for u', v' the (distinct) vertices of G' corresponding to u, v , respectively.*

Before proving that balanced critical dual solutions exist, we give a lemma showing why they are useful. In particular they show how the $M(v)$ values relate to $T(\pi)$. Let $M(G)$ be a minimum weight perfect matching in G .

Lemma 25. *Let $G = (V, E, w, W)$ be an undirected connected graph with every edge in some minimum weight perfect matching. Let π be a balanced critical dual solution for G . For any vertex $z \in V$, a minimum weight almost perfect matching in $G \setminus z$ weighs $w(M(G)) - H(T(\pi)) - \text{dist}_{T(\pi)}(z, V)$.*

Proof. Any almost perfect matching in $G \setminus z$ weighs at least $w(M(G)) - H(T(\pi)) - \text{dist}_{T(\pi)}(z, V)$. In proof let M_1 be an arbitrary perfect matching in $G \setminus \{x, z\}$ for any $x \in V$. For any blossom U of π such that $x, z \notin U$, $|M_1 \cap \delta(U)| \geq 1$. Together with (5) this gives $w(M_1) \geq \sum_{w \in V - x, z} \pi_w + \sum_{x, z \notin U} \pi_U$. The right-hand side equals $w(M(G)) - \pi_x - \pi_z - \sum_{\{x, z\} \cap U \neq \emptyset} \pi_U$, by strong duality. Since every π_U is nonnegative this quantity is at least $w(M(G)) - \text{dist}_{T(\pi)}(z, V) - \text{dist}_{T(\pi)}(x, V)$. The definition of $H(T(\pi))$ shows the last quantity is at least $w(M(G)) - \text{dist}_{T(\pi)}(z, V) - H(T(\pi))$ as desired.

We complete the proof by constructing an almost perfect matching in $G \setminus z$ of weight $w(M(G)) - H(T(\pi)) - \text{dist}_{T(\pi)}(z, V)$. Take G', u, v, u', v' as in Definition 24. Moreover let z' be the vertex of G' corresponding to z . G' is connected, with every edge in a perfect matching, so it satisfies the hypothesis of Lemma 23. Definition 24 shows that $u' \mathcal{R}v'$. So z' is not equivalent to at least of u and v . W.l.o.g. assume that $u' \mathcal{R}z'$. Thus $G' \setminus \{u', z'\}$ has a perfect matching M_0 .

Next, consider each inclusionwise maximal blossom U of π one by one. Let $x \in U$ be the unique vertex of U in the set $V(M_0) \cup \{u, z\}$. Add to M_0 the edges of the matching $M(U, x)$ guaranteed by Lemma 18.

Clearly M_0 is a perfect matching in $G \setminus \{u, z\}$. For each blossom U of π , $|M_0 \cap \delta(U)|$ is 1 if $u, z \notin U$, and 0 if u or z belongs to U . Blossoms of the latter type are those in the path from u to V or z to V in $T(\pi)$. These two paths have disjoint edge sets, since $u' \neq z'$. We get $w(M_0) = w(M(G)) - \text{dist}_{T(\pi)}(u, V) - \text{dist}_{T(\pi)}(z, V)$, since every edge of M_0 is allowed, i.e., tight. Since $\text{dist}_{T(\pi)}(u, V) = H(T(\pi))$ this is the desired weight. \square

We prove that balanced critical duals exist in two steps. The first step shows a simpler property for critical duals actually makes them balanced. The second step shows duals with this property exist.

Lemma 26. *Let $G = (V, E, w, W)$ be an undirected connected graph with every edge in some minimum weight perfect matching. A critical dual π_0 is balanced if it has minimum height (i.e., $H(T(\pi_0))$ is no larger than the height of any other critical dual).*

Proof. Assume for the purpose of contradiction that π_0 is not a balanced critical dual. For any vertex $v \in V$ let h_v denote its height in π_0 , $h_v = \text{dist}_{T(\pi_0)}(v, V)$. Let u be the vertex of G with the greatest height h_u . Let G' be the graph G with inclusionwise maximal blossoms of π_0 contracted. Let R be the equivalence relation of Lemma 23 for G' , and S_1, \dots, S_k its equivalence classes. Let u belong to vertex u' of G' and let $u' \in S_1$.

We will define a dual function π_1 . An element of S_1 is either a maximal blossom of π_0 or a vertex of V not in any blossom; let s_i , $1 \leq i \leq |S_1|$, be the i th of these blossoms and vertices. Lemma 23 shows $G' \setminus S_1$ has $|S_1|$ connected components; let B_i , $1 \leq i \leq |S_1|$, be the set of vertices of G contracted onto the i -th connected component of $G' \setminus S_1$. Define $\pi_1 : \Omega \cup V \rightarrow \mathbb{R}$ to be identical to π_0 except

$$\pi_1(x) = \begin{cases} \pi_0(x) - \epsilon & x = s_i, 1 \leq i \leq |S_1| \\ \pi_0(x) + \epsilon & x = B_i, 1 \leq i \leq |S_1|. \end{cases}$$

(Note that if B_i consists of more than one vertex in G' then we are creating a new blossom.) Let ϵ be any positive real no larger than the smallest value of $\pi_0(s_i)$ for a blossom s_i . This ensures π_1 is nonnegative on blossoms.

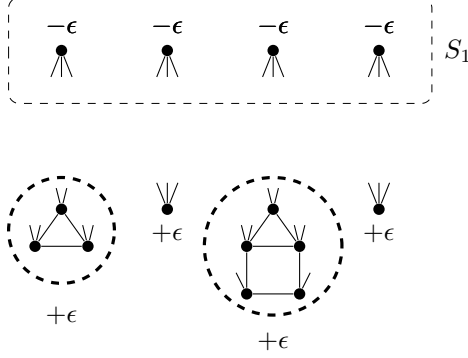


Figure 2: Graph G' and the modification of the duals.

Let us verify that π_1 is a critical dual. First, observe that each edge of G remains tight in π_1 : Nothing changes for an edge that has both ends in the same set of S_1 or some B_i . The remaining possibility is an edge between S_1 and some B_i (no edge joins 2 B_i sets or 2 sets of S_1 , the latter by independence of S_1). For such edges we have added and subtracted ϵ in the left-hand side of 5, so it remains tight. Next observe that the blossoms of π_1 form a laminar family. Lemma 23 shows the sets B_i induce factor critical graphs. Finally π_1 is an optimum dual, since its objective as π_0 . Thus π_1 is a critical dual.

Taking ϵ small enough makes π_1 a critical dual with smaller height than π_0 , the desired contradiction. To see this take any vertex $v \in V$, and let v' be the vertex of G' that v is contracted onto. If $v' \in S_1$, the height of v decreases as long as ϵ is positive. Suppose $v' \notin S_1$. Lemma 23 shows π_0 would be balanced if $h_v = h_u$. Thus $h_v < h_u$. Choose ϵ small enough so that every such v has $\text{dist}_{T(\pi_1)}(v) = \text{dist}_{T(\pi_0)}(v) + \epsilon \leq h_u - \epsilon$. Thus π_1 has smaller height than π_0 . \square

Lemma 27. *Let $G = (V, E, w, W)$ be an undirected connected graph with every edge in some minimum weight perfect matching. There is a critical dual π_0 that has the smallest height $H(T(\pi_0))$.*

Proof. Lemma 16 shows a critical dual π exists. There are a finite number of laminar families on V , i.e., a finite number of trees $T(\pi)$. So it suffices to show that there is a smallest height among all critical duals π with the same tree $T = T(\pi)$.

We begin by showing that for every blossom U , there is a unique value for π_x , where x is any vertex of U or any blossom properly contained in U . We argue inductively, so assume this holds for every blossom properly contained in U . For any edge uv we break the left-hand side of (5) into the contributions from u and from v , by defining

$$\pi_{u,v} = \pi_u + \sum_{U \in \Omega, u = \{u,v\} \cap U} \pi_U$$

and symmetrically for $\pi_{v,u}$. So the left-hand side of (5) is $\pi_{u,v} + \pi_{v,u}$.

Take any edge uv joining two vertices $u, v \in U$. uv is on an odd cycle C contained in U . (U is factor critical, so let M_u (M_v) be a perfect matching on $U - u$ ($U - v$) respectively. The symmetric difference $M_u \oplus M_v$ contains an even-length path from u to v .) Each edge of C is tight. So for every edge xy in C , the values of $\pi_{x,y}$ and $\pi_{y,x}$ are uniquely determined. If $\pi_{x,y}$ does not have any contributions from blossoms properly contained in U then $\pi_x = \pi_{x,y}$ has been uniquely determined. If $\pi_{x,y}$ has

a contribution π_W from a blossom W that is a maximal blossom properly contained in U then π_W has been uniquely determined. This follows since the other π values contributing to $\pi_{x,y}$ have been determined by induction. (Note that π_W has also been uniquely determined from the other edge of $C \cap \delta(W)$.) If neither of these conditions apply to $\pi_{x,y}$ then all its π -values have been determined by induction. Since any vertex $u \in U$ is on an edge uv in U , this completes the inductive argument.

Next consider any edge uv not contained in a blossom of T . The previous argument shows exactly one term in the quantity $\pi_{u,v}$ is still undetermined. If uv is in an odd cycle C the previous argument shows that term is uniquely determined. Contract all such odd cycles as well as all blossoms of T . We get a bipartite graph G' . It contains at least one edge. Let S be a spanning tree of G' . Choose a value p_0 for the unknown term p at the root of S , that comes from a valid critical dual for T . Suppose we increase p . If this increases $H(T)$, every value of p larger than p_0 gives larger height. Suppose this decreases $H(T)$. All the other unknown π -values are uniquely determined from tightness of the edges of S . Also every edge of G' not in S remains tight by bipartiteness. There is a maximum value \bar{p} such that every value $p > \bar{p}$ either makes the π -values invalid (because some $\pi_U, U \in \Omega$ becomes negative) or increases the height (since p contributes to the height of the root vertex). Similarly there is a minimum value \underline{p} for p . We conclude there is a unique smallest height for a critical dual for T – it occurs when p is equal to either \bar{p} or \underline{p} . \square

As already mentioned, the last two lemmas show any undirected connected graph $G = (V, E, w, W)$ with all edges allowed has a balanced critical dual. We can now reach our final goal.

Lemma 28. *Let $G = (V, E, w, W)$ be a weighted undirected connected graph where every edge is allowed. Given all values $w(M(v))$ for $v \in V$, the blossoms of a balanced critical dual solution can be found in $\tilde{O}(n^2)$ time.*

Proof. Let π be a balanced critical dual solution. By Lemma 25 for each leaf node $v \in V$ of $T(\pi)$, $w(M(v)) = w(M(G)) - H(T(\pi)) - \text{dist}_{T(\pi)}(z, V)$. Define new edge weights $w' : E \rightarrow \mathbb{Z}$ as $w'(uv) = w(M(u)) + w(M(v)) + w(uv)$. Consider any $uv \in E$. Since uv is tight, $w(uv) = \text{dist}_{T(\pi)}(u, v)$. Define a quantity c that is independent of uv , $c = 2(w(M(G)) - H(T(\pi)))$. Then

$$\begin{aligned}
w'(uv) &= w(uv) + w(M(u)) + w(M(v)) \\
&= \text{dist}_{T(\pi)}(u, v) + 2w(M(G)) - 2H(T(\pi)) - \text{dist}_{T(\pi)}(u, V) - \text{dist}_{T(\pi)}(v, V) \\
&= 2(w(M(G)) - H(T(\pi))) - 2\text{dist}_{T(\pi)}(\text{lca}(u, v), V) \\
&= c - 2\text{dist}_{T(\pi)}(\text{lca}(u, v), V).
\end{aligned} \tag{6}$$

Let $B = \text{lca}(u, v)$. So B is the inclusionwise minimal blossom of π containing both u and v , or if no such blossom exists, B is the root V of the tree $T(\pi)$. For any edge uv let $B_{uv} \subseteq V$ be the set of vertices reachable from u or v by a path of edges e satisfying $w'(e) \leq w'(uv)$.

Claim. *For any edge uv , $B_{uv} = B$.*

Proof of Claim. Let $F \subseteq E$ be the set of edges of a spanning tree of $G[B]$ ($G[B]$ is connected since either $B = V$ or $G[B]$ is factor critical). Since any edge $ab \in F$ is contained in B , the node $\text{lca}(a, b)$ descends from $\text{lca}(u, v)$ in $T(\pi)$. Thus the path from $\text{lca}(a, b)$ to $\text{lca}(u, v)$ in T has nonnegative weight. This implies $w'(ab) \leq w'(uv)$ by (6). Thus $B \subseteq B_{uv}$.

For the opposite inclusion, consider any edge ab with $a \in B$ and $w'(ab) \leq w'(uv)$. Since every blossom has a strictly positive π -value, (6) implies $b \in B$. Now an easy induction shows any path from u or v , with every edge e having $w'(e) \leq w'(uv)$, has every vertex in B . Thus $B_{uv} \subseteq B$. \diamond

Any blossom B of π has an edge uv with B the minimal blossom containing u and v (by laminarity and connectedness of B). So the claim of the lemma amounts to constructing all the sets B_{uv} . This is done in $\tilde{O}(n^2)$ time by Algorithm 5 below. \square

Algorithm 5 Given all the values $w(M(u))$, finds the blossoms of a balanced critical dual in the graph G where all edges are allowed.

- 1: For each edge uv set $w'(uv) = w(uv) + w(M(u)) + w(M(v))$.
 - 2: Let A be the set of all different values $w'(uv)$. Let $\mathcal{B} = \emptyset$.
 - 3: **for** each $\alpha \in A$, in increasing order, **do**
 - 4: Let \mathcal{C} be the set of connected components of the graph $(V, \{uv : uv \in E, w'(uv) \leq \alpha\})$.
 - 5: Add the nontrivial components of \mathcal{C} to \mathcal{B} .
 - 6: **end for**
 - 7: **return** \mathcal{B} .
-

6.3 The Final Algorithm

Theorem 29. *Let $G = (V, E, w, W)$ be a weighted undirected graph containing a perfect matching. A minimum weight perfect matching in G can be computed $\tilde{O}(Wn^\omega)$ time, with high probability.*

Proof. First, using Corollary 11, we can remove all the edges of G which are not allowed. Clearly, we can consider each connected component of G separately, hence w.l.o.g. we assume that G is connected. Next, compute all the values $w(M(u))$ for each $u \in V$ using Corollary 15. Having all the values $w(M(u))$ by Lemma 28 we can find the set of blossoms \mathcal{B} of a balanced critical dual solution and consequently by Lemma 20 we can find a minimum weight perfect matching in G . \square

The full version of this paper shows how the matching algorithm can be made Las Vegas.

In some applications the second smallest perfect matching is of interest (e.g., Section 9). Its weight is easily found, as follows. As discussed in the proof of Lemma 13, the terms in the determinant of $\det(\tilde{A}(G))$ correspond to even-cycle covers in the graph G . Each such cycle can be decomposed into two perfect matchings. As we already observed the smallest degree term in y in $\det(\tilde{A}(G))$ corresponds to taking twice the minimum weight perfect matching in G . The next smallest term gives the following.

Corollary 30. *Let $G = (V, E, w, W)$ be a weighted undirected graph. The degree in y of a second smallest monomial of $\det(\tilde{A}(G))$ is equal to the weight of a minimum weight perfect matching M^* plus the weight of a second smallest perfect matching M' . In particular, the weight of a second smallest perfect matching can be found in $\tilde{O}(Wn^\omega)$ time, with high probability.*

Proof. Letting $\tau(M^*)$ denote the term corresponding to M^* in $\det(\tilde{A}(G))$,

$$w(M') = \text{term}_y^* \left[\det(\tilde{A}(G)) - \tau(M^*) \right]. \quad \square$$

7 Diameter and Radius

In this section we consider the problem of computing the diameter and radius of a directed graph without negative weight cycles. By bidirecting the edges this result can be applied to undirected graphs with nonnegative edge weights.

We start with definitions of the quantities of interest, plus equivalent definitions that we use to compute these quantities. To motivate the latter note that it seems difficult to compute a given set of distances S directly. Instead we show how to check if, for an arbitrary value c , all the distances in S are $\leq c$. For a graph G and a vertex i ,

$$\begin{aligned} \text{eccentricity}(i) &= \max\{\text{dist}_G(i, j) : j \in V\} &= \min\{c : (\forall j)(c \geq \text{dist}(i, j))\}, \\ \text{radius}(G) &= \min\{\text{eccentricity}(i) : i \in V\} &= \min\{c : (\exists i)(\forall j)(c \geq \text{dist}(i, j))\}, \\ \text{diameter}(G) &= \max\{\text{eccentricity}(i) : i \in V\} &= \min\{c : (\forall i, j)(c \geq \text{dist}(i, j))\}. \end{aligned}$$

We use the following theorem proven in [27]³:

Lemma 31. *Let \vec{G} be a directed weighted graph without negative weight cycles. The weight of the shortest path in G from i to j is given by $\text{dist}_G(i, j) = \deg_y^* \left(\text{adj} \left(\tilde{A}(\vec{G}) + I \right)_{i,j} \right)$. Moreover, all non-zero terms in $\det \left(\tilde{A}(\vec{G}) \right)_{i,j}$ are non-zero over any finite field \mathcal{Z}_p .*

In order to be able to use the above lemma we first need to apply the following observation.

Corollary 32. *Let c be arbitrary number from $[-nW, \dots, nW]$. There exists $d \leq c$ such that $\text{term}_y^d \left[\text{adj} \left(\tilde{A}(\vec{G}) + I \right)_{i,j} \right] \neq 0$ if and only if $\text{term}_y^c \left[\text{adj} \left(\tilde{A}(\vec{G}) + I \right)_{i,j} \cdot \left(\sum_{i=0}^{2nW} y^i \right) \right] \neq 0$. This continues to hold in any finite field.*

Proof. Multiplication of a polynomial by $\sum_{i=0}^{2nW} y^i$ means that we add all lower degree terms to a term of a given degree. Hence, if some degree term d was non-zero then all higher degree terms become non-zero (assuming no terms get cancelled). Since $0 \leq c - d \leq 2nW$ a degree d term creates a corresponding degree c term in the product.

Finally observe that no term drops out because of cancellations⁴: In the matrix $\tilde{A}(\vec{G}) + I$, every nonzero entry has the form $xy^{w(x)}$, where x is the indeterminate $\neq y$ (or 1 in diagonal entries) and w is a function. So every term in $\det(\tilde{A}(\vec{G}) + I)$ has the exponent of y functionally dependent on the remaining variables. This holds for an entry of the adjoint too. Thus a term $(\prod x)y^c$ created in the multiplication of the corollary comes from exactly one term $(\prod x)y^d$ in the adjoint (i.e., $d = \sum w(x)$). So there are no cancellations, in ordinary arithmetic or in \mathcal{Z}_p . \square

To find the diameter, we use this Corollary to perform a binary search for the lowest c such that for all $i, j \in V$,

$$\text{term}_y^c \left[\text{adj} \left(\tilde{A}(\vec{G}) + I \right)_{i,j} \cdot \left(\sum_{i=0}^{2nW} y^i \right) \right] \neq 0.$$

Clearly this c is the diameter. Similarly a binary search for the lowest c where the displayed condition holds for some i with every j gives the radius.

The main problem left is how to check whether $\text{term}_y^d \left[\text{adj} \left(\tilde{A}(\vec{G}) + I \right)_{i,j} \cdot \left(\sum_{i=0}^{2nW} y^i \right) \right] \neq 0$.

We will show how to obtain adj as a partial derivative of \det . We define \tilde{Z} to be a *fully symbolic matrix* of size $n \times n$ as $\tilde{Z}_{i,j} = z_{i,j}$, where $z_{i,j}$ are unique variables for all $i, j \in [1, \dots, n]$. We define σ_z to be an evaluation that assigns 0 to all $z_{i,j}$. Now we are ready to prove the following lemma.

³In [27] the graph was defined to contain self-loops whereas here we add self-loops in the equation by taking $\tilde{A}(\vec{G}) + I$.

⁴We shall apply this principle to other matrices.

Lemma 33.

$$\text{term}_y^d \left[\text{adj} \left(\tilde{A}(\vec{G}) + I \right)_{i,j} \left(\sum_{i=0}^{2nW} y^i \right) \right] = \frac{\partial}{\partial z_{j,i}} \text{term}_y^d \left[\det \left(\tilde{A}(\vec{G}) + I + \tilde{Z} \right) \left(\sum_{i=0}^{2nW} y^i \right) \right] \Big|_{\sigma_z}.$$

Proof. Observe that for the fully symbolic $n \times n$ matrix \tilde{Z} and any $n \times n$ matrix \tilde{A} not involving any variable $z_{i,j}$,

$$\frac{\partial}{\partial z_{j,i}} \det(\tilde{A} + \tilde{Z})|_{\sigma_z} = \text{adj}(\tilde{A})_{i,j}. \quad (7)$$

Thus

$$\begin{aligned} \frac{\partial}{\partial z_{j,i}} \text{term}_y^d \left[\det \left(\tilde{A}(\vec{G}) + I + \tilde{Z} \right) \left(\sum_{i=0}^{2nW} y^i \right) \right] \Big|_{\sigma_z} &= \text{term}_y^d \left[\frac{\partial}{\partial z_{j,i}} \det \left(\tilde{A}(\vec{G}) + I + \tilde{Z} \right) \Big|_{\sigma_z} \left(\sum_{i=0}^{2nW} y^i \right) \right] \\ &= \text{term}_y^d \left[\text{adj} \left(\tilde{A}(\vec{G}) + I + \tilde{Z} \right)_{i,j} \Big|_{\sigma_z} \left(\sum_{i=0}^{2nW} y^i \right) \right] = \text{term}_y^d \left[\text{adj} \left(\tilde{A}(\vec{G}) + I \right)_{i,j} \left(\sum_{i=0}^{2nW} y^i \right) \right]. \end{aligned}$$

□

Joining the above results together with Theorem 9 and Corollary 32 we get the following algorithm.

Algorithm 6 Checks whether diameter of the directed graph \vec{G} is $\leq c$.

- 1: Let $\partial_z f$ be the routine given by the Baur-Strassen theorem to compute the matrix of partial derivatives $\frac{\partial}{\partial z_{i,j}} \text{term}_y^{c+nW} \left[\left(\sum_{i=0}^{2nW} y^i \right) \det \left((\tilde{A}(\vec{G}) + I + \tilde{Z})y^W \right) \right]$.
 - 2: Generate a random substitution $\sigma : X \rightarrow Z_p$ for a prime p of order $\Theta(n^4)$. Extend it to $\sigma : X \cup Z \rightarrow Z_p$ by setting $\sigma|_Z = \sigma_z$.
 - 3: Compute the matrix $\delta = \partial_z f|_\sigma$.
 - 4: Return true if $\delta_{i,j}$ is non-zero for all $i, j \in V$.
-

Note that the polynomial multiplication in Step 1 need only compute the coefficient of y^{c+nW} and so only uses time $O(nW)$. Using binary search with the above algorithm, and using a similar algorithm for radius, we obtain:

Theorem 34. Let $\vec{G} = (V, E, w, W)$ be a weighted directed graph without negative weight cycles. The diameter and radius of G can be computed in $\tilde{O}(Wn^\omega)$ time, with high probability.

8 Shortest Cycles in Undirected Graphs

Let $G = (V, E, w, W)$ a weighted undirected graph. In this section we consider only the problem of computing shortest cycles when all edges have non-negative weights. The more general case is solved in the next section, and uses the ideas introduced here. Let us define a *symbolic adjacency matrix* of the weighted undirected graph G to be the symbolic matrix polynomial $\tilde{A}(G)$ equal to $\tilde{A}(\vec{G})$, where \vec{G} is the bidirection of G .

Lemma 35. *Let G be an undirected weighted graph with no negative edges, and $d \in [0, nW]$. Some $uv \in E$ has $\left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] \text{term}_y^d \left[\det(\tilde{A}(G) + I) - 1 \right] \neq 0$ iff some cycle packing of weight d contains a component that is an oriented cycle through uv in G . Moreover, for $p \geq 2$, all non-zero terms in the above expression are non-zero over the finite field Z_p .*

Proof. As argued in the proof of Lemma 4 $\det(\tilde{A}(\vec{G}) + I) - 1$ contains only terms that correspond to cycle packings. Moreover, the degree of each non-zero term is equal to the total weight of the cycles in the packing. However, because $\tilde{A}(G)$ was constructed using bidirected graph there might be terms containing both antiparallel edges, that correspond to bidirected edges in the undirected graph.

Next we show that a cycle packing \mathcal{C} in G contributes to the expression of the lemma iff it contains exactly one of the variables $x_{u,v}$ and $x_{v,u}$, and hence contains a simple cycle passing through uv . Moreover we show that in such a case the contribution of \mathcal{C} is a product of variables corresponding to \mathcal{C} and hence the contribution of \mathcal{C} is not cancelled out by a different cycle packing. If \mathcal{C} does not contain $x_{u,v}$ or $x_{v,u}$, clearly it has zero contribution. This leaves two possibilities:

Case 1. \mathcal{C} contains a cycle u, v, u Since $\left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] x_{u,v} x_{v,u} = x_{u,v} x_{v,u} - x_{v,u} x_{u,v} = 0$, \mathcal{C} 's term makes no contribution.

Case 2. \mathcal{C} contains a simple cycle C containing uv The corresponding term contains exactly one of $x_{u,v}$ and $x_{v,u}$ say $x_{u,v}$. We have $\left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] x_{u,v} = x_{u,v}$. Hence, the derivative for this term is nonzero and is equal to the sign of permutation multiplied by the product of the variables of the oriented edges of \mathcal{C} . \square

Similarly as in undirected graphs we say that edge e is *allowed* if and only if it belongs to some simple shortest cycle C in G . The above proof actually gives us a way to find allowed edges as well.

Corollary 36. *Let d be smallest number in $[1, nW]$ such that there exists an edge $uv \in E$ such that $\left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] \text{term}_y^d \left[\det(\tilde{A}(G) + I) - 1 \right] \neq 0$. Then an edge $uv \in E$ is allowed if and only if when $\left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] \text{term}_y^d \left[\det(\tilde{A}(G) + I) - 1 \right] \neq 0$.*

Computing the Weight of the Shortest Cycle Using Lemma 35 we will devise an algorithm that will be able to check whether there exists a simple cycle in G of length shorter or equal to c . In order to do it we need the observation similar to Corollary 32.

Corollary 37. *Let c be arbitrary number from $[1, nW]$. There exists $d \leq c$ such that*

$$\left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] \text{term}_y^d \left[\det(\tilde{A}(G) + I) - 1 \right] \neq 0$$

if and only if $\left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] \text{term}_y^c \left[(\det(\tilde{A}(G) + I) - 1) \cdot (\sum_{i=0}^{nW} y^i) \right] \neq 0$.

Using the above observation we can construct the following algorithm.

Algorithm 7 Checks whether the shortest cycle in undirected graph G has weight $\leq c$.

- 1: Let $\partial_x f$ be the routine given by the Baur-Strassen theorem to compute the matrix of partial derivatives $\frac{\partial}{\partial x_{u,v}} \text{term}_y^c \left[\left(\sum_{i=0}^{nW} y^i \right) \left(\det(\tilde{A}(G) + I) - 1 \right) \right]$.
 - 2: Generate a random substitution $\sigma : X \rightarrow Z_p$ for a prime p of order $\Theta(n^4)$.
 - 3: Compute the matrix $\delta = \partial_x f|_\sigma$.
 - 4: Compute the matrix δ' with $\delta'_{u,v} = [x_{u,v}\delta_{u,v} - x_{v,u}\delta_{v,u}]|_\sigma$.
 - 5: Return true if δ' has a non-zero entry.
-

The correctness of the above algorithm is implied by both Theorem 8 and Theorem 9. Using binary search with it we obtain.

Theorem 38. *Let $G = (V, E, w, W)$ be a weighted undirected graph without negative weight edges. The weight of the shortest simple cycle in G can be computed in $\tilde{O}(Wn^\omega)$ time, with high probability.*

Finding the Shortest Cycle After showing how to compute the shortest cycle length it remains to show how to find the cycle itself. We essentially can use the same approach as we used for directed graphs in Section 4.

Algorithm 8 Computes the shortest cycle in undirected graph G .

- 1: Let c^* be the weight of the shortest cycle computed using Theorem 38.
 - 2: Let δ' be the matrix computed by Algorithm 7 for $c = c^*$.
 - 3: Take any edge uv such that $\delta_{uv} \neq 0$.
 - 4: Compute the shortest path $p_{v,u}$ from v to u in $G \setminus \{uv\}$ using Dijkstra.
 - 5: Return the cycle formed by uv and $p_{v,u}$.
-

Theorem 39. *Let $G = (V, E, w, W)$ be a weighted undirected graph without negative weight edges. The shortest simple cycle in G can be found in $\tilde{O}(Wn^\omega)$ time, with high probability.*

9 Undirected Graphs with Negative Weights

This section gives algorithms for shortest cycle and diameter in undirected graphs with possibly negative edges but no negative weight cycles. To accomplish this we need to combine the ideas from Sections 7 and 8 with our results for weighted matching. We will recast the results for the diameter and shortest cycles into the language of matchings. Hence (unlike Section 8) throughout this section the symbolic adjacency matrix $\tilde{A}(G)$ of an undirected graph G is defined as in Section 6.1.

Diameter Let $G = (V, E, w, W)$ be an undirected graph with negative weights allowed, and let E^- be the set of edges with negative weights. We will define a graph \tilde{G} that models paths in G by almost perfect matchings. We believe the construction is essentially due to Edmonds [6].

Define the *split graph* $\tilde{G} = (\tilde{V}, \tilde{E}, \tilde{w}, W)$ where

$$\begin{aligned} \tilde{V} &= \{v_1, v_2 : v \in V\} \cup \{e_1, e_2 : e \in E^-\}, \\ \tilde{E} &= \{v_1 v_2 : v \in V\} \cup \{u_1 v_2, u_2 v_1, u_1 v_1, u_2 v_2 : uv \in E \setminus E^-\} \\ &\quad \cup \{u_1 e_1, u_2 e_1, e_1 e_2, v_1 e_2, v_2 e_2 : e = uv \in E^-, u < v\}, \end{aligned}$$

$$\ddot{w}(u_i v_j) = \begin{cases} w(uv) & \text{if } uv \in E \setminus E^-, \\ w(e) & \text{if } u_i = e_1 \text{ and } v_j \neq e_2 \text{ and } e \in E^-, \\ 0 & \text{otherwise.} \end{cases}$$

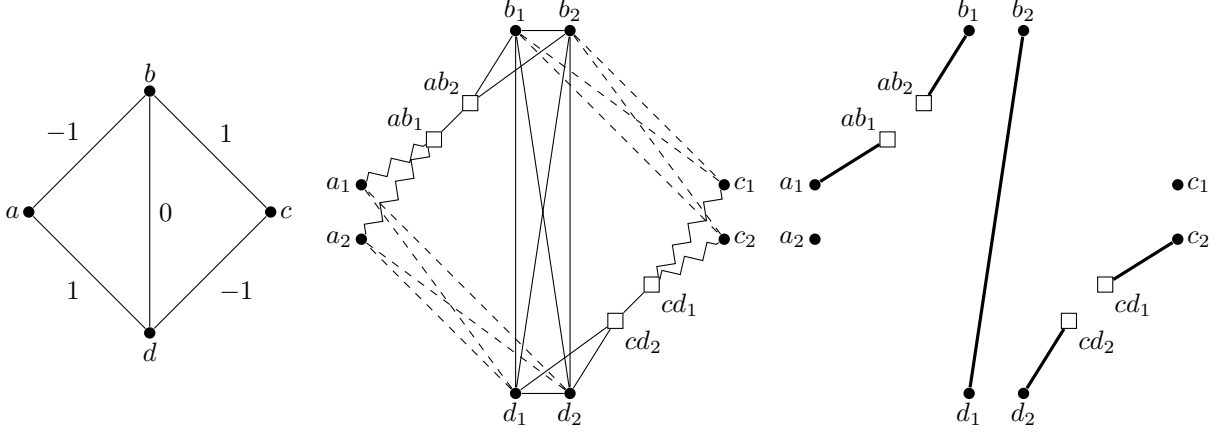


Figure 3: An undirected graph G and its graph \ddot{G} . In \ddot{G} zigzag edges weigh -1 , dashed edges weigh 1 and the remaining edges weigh 0 . Vertices corresponding to negative edges of G are white squares. The far right shows a matching $M(a_2c_1)$ of weight -2 , which corresponds to a shortest path between a and c .

Note how a length-two path in G , say a, b, c with $w(ab) \geq 0 > w(bc)$ and $e = bc$, corresponds to a matching in \ddot{G} such as a_1b_1, b_2c_1, e_2c_1 , having the same total weight. Fig.3 gives a complete example.

An important property is that we can assume $\ddot{n} = |\ddot{V}| \leq 4n$. This follows since we can assume $|E^-| < n$, as otherwise the set of negative edges contains a cycle.

We will consider minimum weight perfect matchings in \ddot{G} . To use our algebraic tools we should eliminate negative weights by setting $w'(e) := \ddot{w}(e) + W$. Obviously this increases the weight of all perfect matchings by $\ddot{n}W/2$ and so doesn't change the minimum perfect matching. But to keep things simple in the following we keep \ddot{G} as defined above, with \ddot{w} possibly negative.

The following observation is essentially given in [1] in Chapter 12.7 (for a larger version of our graph).

Lemma 40. *Let $u, v \in V$, let M be the minimum weight perfect matching, and let $M(u_2v_1)$ be the minimum weight almost perfect matching in \ddot{G} that does not match v_1 nor u_2 . If G does not contain negative weight cycles then $\ddot{w}(M) = 0$ and the shortest path weight from u to v in G is equal to $\ddot{w}(M(u_2v_1))$.*

Note also that it is easy to detect a negative cycle in G – it corresponds to a perfect matching in \ddot{G} with negative weight.

By Lemma 13 we know that $\deg_y^*(\text{adj}(\tilde{A}(\ddot{G}))_{u_2, v_1}) = \ddot{w}(M) + \ddot{w}(M(u_2v_1))$. Thus $\text{dist}_G(u, v) = \deg_y^*(\text{adj}(\tilde{A}(\ddot{G}))_{u_2, v_1})$, i.e., just as in Lemma 31, $\text{adj}(\tilde{A}(\ddot{G}))$ encodes the distances in G . Now we proceed exactly as in Algorithm 6.

Algorithm 9 Checks whether diameter of the undirected graph G with negative weights is $\leq c$.

- 1: Let $\partial_z f$ be the routine given by the Baur-Strassen theorem to compute the matrix of partial derivatives $\frac{\partial}{\partial z_{u_2, v_1}} \text{term}_y^{c+\tilde{n}W} \left[\left(\sum_{i=0}^{2nW} y^i \right) \det \left((\tilde{A}(\ddot{G}) + \tilde{Z})y^W \right) \right], u, v \in V$.
 - 2: Generate a random substitution $\sigma : X \rightarrow Z_p$ for a prime p of order $\Theta(n^4)$. Extend it to $\sigma : X \cup Z \rightarrow Z_p$ by setting $\sigma|_Z = \sigma_z$.
 - 3: Compute the matrix $\delta = \partial_z f|_\sigma$.
 - 4: Return true if δ_{u_2, v_1} is non-zero for all $u, v \in V$.
-

Note that $\tilde{A}(\ddot{G}) + \tilde{Z}$ is not skew-symmetric but we still get the adjoint by (7). Similarly to check if the radius is $\leq c$, Step 4 returns true if some row of δ consists entirely of nonzeros. Again using binary search we obtain.

Theorem 41. *Let $G = (V, E, w, W)$ be a weighted undirected graph without negative weight cycles. The diameter and radius of G can be computed in $\tilde{O}(Wn^\omega)$ time, with high probability.*

Shortest Cycles Recalling Corollary 30, it might appear that a second smallest perfect matching in \ddot{G} corresponds to a shortest cycle in G . But this is not true, because of cycles of length two (e.g., u_1v_1, v_2u_2 or u_2v_1, v_2u_1). These can be handled as in Section 8, by antisymmetric derivatives $x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}}$. For the next lemma note that in the absence of negative cycles, any cycle contains an edge of nonnegative weight.

Lemma 42. *Let G be an undirected weighted graph with no negative cycle. For any edge $uv \in E \setminus E^-$, a shortest cycle through uv weighs $\text{term}_y^* \left[x_{u_1, v_2} \frac{\partial}{\partial x_{u_1, v_2}} - x_{u_2, v_1} \frac{\partial}{\partial x_{u_2, v_1}} \right] \left(\det(\tilde{A}(\ddot{G})) \right)$. This continues to hold in any field Z_p , $p \geq 2$.*

Proof. Let $\delta = \left[x_{u_1, v_2} \frac{\partial}{\partial x_{u_1, v_2}} - x_{u_2, v_1} \frac{\partial}{\partial x_{u_2, v_1}} \right] \left(\det(\tilde{A}(\ddot{G})) \right)$. Observe that in general for any variables x, y and any integers i, j ,

$$\left[x \frac{\partial}{\partial x} - y \frac{\partial}{\partial y} \right] x^i y^j = (i - j) x^i y^j. \quad (8)$$

Hence the terms in δ are a subset of those in $\det(\tilde{A}(\ddot{G}))$, i.e., the effect of the differentiation operator is just to change the multiplicity of some terms, perhaps zeroing them or causing other cancellations.

Let C be a shortest cycle through uv . It gives an even-cycle cover $M^* \cup N$ in \ddot{G} with weight $w(C)$, where M^* is the minimum perfect matching of \ddot{G} , $M^* = \{v_1v_2, e_1e_2 : v \in V, e \in E^-\}$, and N is the perfect matching of \ddot{G} containing u_1v_2 but not u_2v_1 , plus representatives of the other edges of C , plus edges x_1x_2 for vertices or negative edges $x \notin C$. Let τ be the monomial corresponding to this cover (e.g., τ has the term $y^{w(C)}$). (8) (applied to $x_{u_1, v_2}^1 x_{u_2, v_1}^0$) shows τ is a term contributing to δ . In fact τ is the only such term in δ involving its monomial. This again follows from (8) and the preliminary observation (it is easy to see τ is the unique edge cover for its monomial, i.e., none of its cycles can be reversed). We conclude $\text{term}_y^*(\delta) \leq w(C)$.

We complete the proof by showing $\text{term}_y^*(\delta) \geq w(C)$. First observe that a perfect matching M on \ddot{G} that contains u_1v_2 but not u_2v_1 weighs at least $w(C)$. To prove this imagine contracting each edge x_1x_2 of \ddot{G} ; call the resulting vertex x . Every vertex now has degree 2 in M . We will compute the weight of M by examining its edges in the contracted graph. An edge $x_1x_2 \in M$ becomes a

loop at x of weight 0. An edge $xy \in E \setminus E^-$ with $x_1y_1, x_2y_2 \in M$ or $x_1y_2, x_2y_1 \in M$ becomes 2 copies of xy , both with nonnegative weight. The other edges of M form cycles in the contracted graph. Each cycle is a cycle in G and so has nonnegative weight. One of the cycles contains edge uv , so it weighs at least $w(C)$. Hence $\check{w}(M) \geq w(C)$.

Consider an even-cycle cover \mathcal{C} that contributes to δ . (8) shows $i \neq j$, i.e., u_1v_2 and u_2v_1 occur with different multiplicities in \mathcal{C} . The possibilities for $\{i, j\}$ are $\{0, 1\}$, $\{0, 2\}$, and $\{1, 2\}$. \mathcal{C} decomposes into 2 perfect matchings. In all three cases one of the matchings of \mathcal{C} contains exactly 1 of the edges u_1v_2, u_2v_1 . That matching weighs at least $w(C)$. The other matching has nonnegative weight, so \mathcal{C} weighs at least $w(C)$. In other words $\text{term}_y^*(\delta) \geq w(C)$. \square

Using this lemma with the scheme of Algorithm 7 gives the following.

Algorithm 10 Checks whether a shortest cycle in undirected graph G with negative weights has weight $\leq c$.

- 1: For $(i, j) = (1, 2), (2, 1)$, let $\partial_x f^{i,j}$ be the routine given by the Baur-Strassen theorem to compute the matrix of partial derivatives $\frac{\partial}{\partial x_{u_i, v_j}} \text{term}_y^{c+\check{n}W} \left[\left(\sum_{i=0}^{\check{n}W} y^i \right) \left(\det(\tilde{A}(\check{G})y^W) \right) \right]$.
 - 2: Generate a random substitution $\sigma : X \rightarrow Z_p$ for a prime p of order $\Theta(n^4)$.
 - 3: For $(i, j) = (1, 2), (2, 1)$ compute the matrix $\delta^{i,j} = \partial_x f^{i,j}|_\sigma$.
 - 4: Compute the matrix δ' with $\delta'_{u,v} = \left[x_{u_1, v_2} \delta_{u,v}^{1,2} - x_{u_2, v_1} \delta_{u,v}^{2,1} \right] \Big|_\sigma$.
 - 5: Return true if δ' has a non-zero entry.
-

Again a binary search gives

Theorem 43. *Let $G = (V, E, w, W)$ be a weighted undirected graph without negative weight cycles. The weight of a shortest cycle can be computed in $\tilde{O}(Wn^\omega)$ time, with high probability.*

Algorithm 10 with c equal to the shortest cycle weight gives an edge uv with $\delta'_{u,v} \neq 0$, i.e., uv is on a shortest cycle. So a minimum weight perfect matching on $\check{G} - u_1, v_2$ corresponds to a shortest cycle. Hence we can state

Theorem 44. *Let $G = (V, E, w, W)$ be a weighted undirected graph without negative weight cycles. A shortest cycle in G can be found in $\tilde{O}(Wn^\omega)$ time, with high probability. The same holds for a shortest st -path, for any given vertices s, t .*

10 Vertices Lying on Short Cycles

For undirected graphs we only need to change the output of our algorithms: For Algorithm 7 (when there are no negative edges) or Algorithm 10 (in the general case) we find the set of vertices lying on cycles of length $\leq c$ by changing the last step so that it returns $\{v \in V : \exists_{vu \in E} \delta'_{vu} \neq 0\}$.

For directed graphs we apply the $\sum_{i=0}^{2nW} y^i$ multiplication technique to Algorithm 2:

Algorithm 11 Computes the set of vertices lying on cycles of length $\leq c$ in directed graphs.

- 1: Let $\partial_x f$ be the routine given by the Baur-Strassen theorem to compute the matrix of partial derivatives $\frac{\partial}{\partial x_{u,v}} \text{term}_y^{c+nW} \left[\left(\sum_{i=0}^{2nW} y^i \right) \left(\det((\vec{A}(\vec{G}) + I)y^W) - y^{nW} \right) \right]$.
 - 2: Generate a random substitution $\sigma : X \rightarrow Z_p$ for a prime p of order $\Theta(n^4)$.
 - 3: Compute the matrix $\delta = \partial_x f|_\sigma$.
 - 4: Return $\{v \in V : \exists_{(v,u) \in E} \delta_{(v,u)} \neq 0\}$.
-

Thus we obtain:

Theorem 45. *Let G be a weighted directed or undirected graph with integral weights in $[-W, W]$ and no negative cycle. For any c the set of vertices lying on cycles of length $\leq c$ can be computed in $\tilde{O}(Wn^\omega)$ time, with high probability.*

References

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [2] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22(3):317–330, 1983.
- [3] A. Björklund. Determinant sums for undirected hamiltonicity. In *Prof. of FOCS'10*, pages 173–182, 2010.
- [4] T. M. Chan. More algorithms for all-pairs shortest paths in weighted graphs. In *Proc. of STOC'07*, pages 590–598, 2007.
- [5] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research National Bureau of Standards-B.*, 69B:125–130, 1965.
- [6] J. Edmonds. An introduction to matching. Mimeographed notes, Engineering Summer Conference, U. Michigan, Ann Arbor, MI, 1967.
- [7] H. N. Gabow. An efficient implementation of edmonds' algorithm for maximum matching on graphs. *Journal of the ACM*, 23(2):221–234, 1976.
- [8] H. N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proc. of STOC'83*, pages 448–456, 1983.
- [9] H. N. Gabow. A scaling algorithm for weighted matching on general graphs. In *Proc. of FOCS'85*, pages 90–100, 1985.
- [10] H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proc. of SODA'90*, pages 434–443, 1990.
- [11] H. N. Gabow, Z. Galil, and T. Spencer. Efficient implementation of graph algorithms using contraction. *Journal of the ACM*, 36(3):540–572, 1989.

- [12] H. N. Gabow and R. E. Tarjan. Faster scaling algorithms for general graph matching problems. *Journal of the ACM*, 38(4):815–853, 1991.
- [13] Z. Galil, S. Micali, and H. N. Gabow. An $O(EV \log V)$ algorithm for finding a maximal weighted matching in general graphs. *SIAM Journal on Computing*, 15(1):120–130, 1986.
- [14] N. J. A. Harvey. Algebraic structures and algorithms for matching and matroid problems. In *Proc. of FOCS'06*, pages 531–542, 2006.
- [15] C.-C. Huang and T. Kavitha. Efficient algorithms for maximum weight matchings in general graphs with small edge weights. In *Proc. of SODA'12*, pages 1400–1412, 2012.
- [16] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. In *Proc. of STOC'77*, pages 1–10, 1977.
- [17] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM*, 24(1):1–13, Jan. 1977.
- [18] M.-Y. Kao, T.-W. Lam, W.-K. Sung, and H.-F. Ting. A decomposition theorem for maximum weight bipartite matchings with applications to evolutionary trees. In *Proc. of ESA'99*, pages 438–449, 1999.
- [19] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986.
- [20] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart, and Winston, New York, New York, 1976.
- [21] L. Lovász and M. D. Plummer. *Matching Theory*. Akadémiai Kiadó, 1986.
- [22] J. Morgenstern. How to compute fast a function and all its derivatives: a variation on the theorem of Baur-strassen. *SIGACT News*, 16(4):60–62, Apr. 1985.
- [23] M. Mucha and P. Sankowski. Maximum matchings via Gaussian elimination. In *Proc. of FOCS'04*, pages 248–255, 2004.
- [24] S. Pettie. A new approach to all-pairs shortest paths on real-weighted graphs. *Theoretical Computer Science*, 312(1):47–74, 2004.
- [25] L. Roditty and V. V. Williams. Minimum weight cycles and triangles: Equivalences and algorithms. In *Proc. of FOCS'11*, pages 180–189, 2011.
- [26] P. Sankowski. Processor efficient parallel matching. In *Proc. of SPAA'05*, pages 165–170, 2005.
- [27] P. Sankowski. Shortest paths in matrix multiplication time. In *Proc. of ESA'05*, pages 770–778, 2005.
- [28] P. Sankowski. Maximum weight bipartite matching in matrix multiplication time. *Theoretical Computer Science*, 410(44):4480–4488, 2009.
- [29] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer-Verlag, 2003.

- [30] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27:701–717, 1980.
- [31] A. Shoshan and U. Zwick. All pairs shortest paths in undirected graphs with integer weights. In *Proc. of FOCS'99*, pages 605–614, 1999.
- [32] A. Storjohann. High-order lifting and integrality certification. *Journal of Symbolic Computation*, 36(3-4):613–648, 2003.
- [33] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984.
- [34] R. Yuster. A shortest cycle for each vertex of a graph. *Information Processing Letters*, 111(21-22):1057–1061, Nov. 2011.
- [35] R. Yuster and U. Zwick. Answering distance queries in directed graphs using fast matrix multiplication. In *Proc. of FOCS'05*, pages 389–396, 2005.
- [36] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proc. of EUROSAM'79*, pages 216–226, 1979.
- [37] U. Zwick. All pairs shortest paths using bridging sets and rectangular matrix multiplication. *Journal of the ACM*, 49(3):289–317, 2002.