

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

Marek Cygan*, Harold N. Gabow†, Piotr Sankowski‡§

*IDSIA, University of Lugano, Switzerland. Email: marek@idsia.ch

†Department of Computer Science, University of Colorado at Boulder, USA. Email: hal@cs.colorado.edu

‡University of Warsaw and Department of Computer and System Science, Poland. Email: sank@mimuw.edu.pl

§Sapienza University of Rome, Italy.

Abstract—Consider a directed or undirected graph with integral edge weights in $[-W, W]$. This paper introduces a general framework for solving problems on such graphs using matrix multiplication. The framework is based on the Baur-Strassen Theorem and Strojohann's determinant algorithm.

For directed and undirected graphs without negative cycles we obtain simple $\tilde{O}(Wn^\omega)$ running time algorithms for finding a shortest cycle, computing the diameter or radius, and detecting a negative weight cycle. For each of these problems we unify and extend the class of graphs for which $\tilde{O}(Wn^\omega)$ time algorithms are known. In particular no such algorithms were known for any of these problems in undirected graphs with (potentially) negative weights.

We also present an $\tilde{O}(Wn^\omega)$ time algorithm for minimum weight perfect matching. This resolves an open problem posed by Sankowski in 2006, who presented such an algorithm for bipartite graphs. Our algorithm uses a novel combinatorial interpretation of the linear program dual for minimum perfect matching.

We believe this framework will find applications for finding larger spectra of related problems. As an example we give a simple $\tilde{O}(Wn^\omega)$ time algorithm to find all the vertices that lie on cycles of length at most t , for given t . This improves an $\tilde{O}(Wn^\omega t)$ time algorithm of Yuster.

Keywords—shortest cycles; diameter; radius; minimum weight perfect matchings; matrix multiplication

I. INTRODUCTION

The application of matrix multiplication to graph problems has been actively studied in recent years. Unweighted graphs are well understood. For example, $\tilde{O}(n^\omega)$ time algorithms for finding shortest cycles [2] have been known for 35 years¹. But similar results for weighted graphs were obtained only last year, by Roditty and Vassilevska-Williams [3]. 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

The full version of this work can be found at [1]. The first author is partially supported by the ERC grant NEWNET, reference 279352 and Foundation for Polish Science. The third author is partially supported by the ERC Starting Grant PAAI 259515.

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.

framework that gives simple² solutions to all three of these problems and others. We obtain the following results. All graphs are allowed to contain negative weights, with negative cycles disallowed in all problems except matching.

Shortest Cycles: We give simple $\tilde{O}(Wn^\omega)$ time algorithms for finding shortest cycles in undirected and directed graphs. For directed graphs the algorithm reduces the problem to computing one determinant of a polynomial matrix. The undirected case requires handling 2-edge cycles properly. Here we extend the algorithm of Sankowski [4] that tests for a negative weight cycle. For directed graphs (and undirected graphs with non-negative weights) our bounds match those obtained in 2011 by Roditty and Vassilevska-Williams [3], whereas for undirected graphs with negative weights no $\tilde{O}(Wn^\omega)$ time algorithm was previously known. For those graphs the problem is known to reduce to finding n minimum weight perfect matchings [5]. Our shortest path algorithms also detect existence of a negative weight cycle.

Diameter and Radius: We present simple $\tilde{O}(Wn^\omega)$ time algorithms for computing the diameter and radius of undirected and directed graphs. The algorithms combine determinant computations with binary search. Since computing all-pairs shortest paths suffices to find diameter and radius, an $\tilde{O}(Wn^\omega)$ time algorithm follows from [6] for undirected graphs with non-negative weights. Using random sampling with [6], one can obtain the same running time for directed graphs without negative cycles. However, to the best of our knowledge, all previous solutions for undirected graphs with negative weights reduced the problem to finding n minimum weight perfect matchings.

Minimum Weight Perfect Matching: We present an $\tilde{O}(Wn^\omega)$ time algorithm for finding a minimum weight perfect matching. This resolves an open problem posed by Sankowski in 2006 [7], who gave such an algorithm for bipartite graphs. Huang and Kavitha [8] recently gave an $\tilde{O}(Wn^\omega)$ time algorithm for maximum weight matching. But minimum weight perfect matching (even in bipartite graphs) requires more structured techniques. Our algorithm

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.

uses a novel combinatorial interpretation of the linear program dual problem, which may be of independent interest.

A. The Framework

Our framework is based on two seminal results. We use Storjohann’s algorithm [9] 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. But the determinant itself does not provide enough information. Here we use the Baur-Strassen Theorem [10], which shows how to compute all partial derivatives of a function in the same asymptotic time as the function itself. ([11] gives a simple constructive proof.) This 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.

We believe this framework will find applications to other problems. Here we give one more illustration: an $\tilde{O}(Wn^\omega)$ time algorithm finding every vertex that lies on a cycle of weight $\leq t$, for a given arbitrary t (for a directed or undirected graph with negative edges allowed but no negative cycles). This improves the time bound of [12] (which does not allow negative edges).

The paper is organized as follows. Sections II–III give the needed tools and main definitions. Section IV gives a simple example of our framework: a shortest cycle algorithm for undirected graphs with non-negative weights. Section V gives the minimum weight perfect matching algorithm. Section VI summarizes ideas used in the full version to find shortest cycles, radius and diameter, detect negative cycles and find the vertices on cycles of length $\leq t$. Due to space limitations, proofs of theorems marked with a spade symbol (♠) are postponed to the full version of this paper.

II. PRELIMINARIES

Linear Algebra Algorithms: Storjohann [9] 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 II.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 [9]),*
- *determinant $\det(A)$ (Algorithm 12 [9]),*

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

For our applications note that these algorithms can be written as straight-line programs. The randomization does not pose a problem as it is just used at the start to generate

a polynomial of degree d not dividing $\det(A)$. Our applications will work over a finite field (Corollary II.3) so there is no risk of manipulating huge integers. Usage of the FFT to multiply degree nd polynomials does not pose a problem.

Baur-Strassen Theorem: Another astonishing result is the Baur-Strassen Theorem from 1983 [10], [11]. It was used to show that matrix multiplication is no harder than determinant computation (for straight-line programs). This is unexpected since matrix multiplication returns n^2 numbers and determinant computation just one. However one can 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 II.2 (Baur-Strassen ’83). *For straight-line programs computing $f(x_1, \dots, x_n)$,*

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

Thus we can find all partial derivatives of f in the same asymptotic time as f . [11] shows a RAM routine to compute all n partials can be constructed in time $O(T(f))$ as well.

Schwartz-Zippel Lemma: We encode a graph problem in a symbolic matrix whose determinant is (symbolically) non-zero iff the problem has a solution. The Schwartz-Zippel Lemma [13], [14] provides the non-zero test:

Corollary II.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.

We note that multivariate determinants have been used in previous work such as Kirchhoff’s Matrix-Tree Theorem, results of Tutte and Edmonds on perfect matchings, and recently undirected Hamiltonicity detection [15].

III. DEFINITIONS: GRAPHS WITH INTEGRAL WEIGHTS

A *weighted n -vertex graph* is a tuple $G = (V, E, w, W)$, where the vertex set is $V = \{1, \dots, n\}$, $E \subseteq V \times V$ is the edge set, and $w : E \rightarrow [-W, W]$ ascribes a weight to each edge. We consider both undirected and directed graphs (so E can consist of unordered pairs or ordered pairs). The *weight of edge set $F \subseteq E$* is $w(F) = \sum_{e \in F} w(e)$.

Consider a walk $p = v_1, v_2, \dots, v_k$. p is a *path* if all the vertices v_i are distinct. A closed walk, i.e. $v_k = v_1$, with the first $k - 1$ vertices distinct is a *cycle*, with the exception of walks v_1, v_2, v_1 in undirected graphs, which are not cycles.

The *weight $w(p)$* of p is the weight of the edge set of p . The *distance $\text{dist}_G(v, w)$* from v to w in G is the minimum weight of a path from v to w . A path from v to w with minimum weight is a *shortest path*.

The *shortest cycle problem* calls for the shortest (i.e., minimum weight) cycle in a given weighted graph (which

contains no negative cycles). The standard approach of reducing an undirected graph G to a directed graph B by bidirecting the edges does not work: B contains length 2 cycles coming from the same undirected edge. Such cycles can be spurious shortest cycles, or even negative cycles.

The *diameter problem* asks for vertices $v, w \in V$ maximizing $\text{dist}_G(v, w)$. Here the reduction of undirected graphs to directed works when the edge weights are non-negative.

A *matching* in an undirected graph is a set of edges having at most one edge incident to each vertex. A *perfect matching* has exactly one edge incident to each vertex. A *minimum weight perfect matching* is a perfect matching M in a weighted graph minimizing $w(M)$. Many other notions of "optimum weighted matching" reduce to minimum weight perfect matching: A *maximum weight perfect matching* is 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 for time bounds linear in W this reduction is not of interest.

IV. SHORTEST CYCLES IN UNDIRECTED GRAPHS

This section illustrates our framework by showing how to find a shortest cycle in an undirected graph with no negative edges in $\tilde{O}(Wn^\omega)$ time. This bound was first proved in [3] using different techniques.

A *symbolic polynomial* $\tilde{p}[y]$ is a multivariate polynomial over a set of variables $y \cup X$, over an arbitrary field K . $\tilde{K}[y] = K[y \cup X]$ denotes the set of symbolic polynomials, and \tilde{K} denotes the set of multivariate polynomials $K[X]$. For a symbolic polynomial \tilde{p} define:

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

If \tilde{p} is the 0 polynomial, $\deg_y^*(\tilde{p}) := \infty$. Call $\sigma : X \rightarrow K$ an *evaluation function*. Define $\tilde{p}[y]|_\sigma$ to be a one variable polynomial in y with all variables $x \in X$ substituted by $\sigma(x)$. Our algorithms will use $K := Z_p$, i.e., we work over a finite field of order p for some prime number p .

A *symbolic polynomial matrix* $\tilde{A}[y] \in \tilde{K}[y]^{n \times n}$ 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.

Consider a weighted undirected graph $G = (V, E, w, W)$. Let \vec{G} be the bidirected version of G (i.e., each edge of G is oriented in both directions in \vec{G}). The (bidirected) *symbolic adjacency matrix* $\tilde{A}(G)$ of G is defined by

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

where $x_{i,j}$ are unique variables corresponding to the edges of \vec{G} (so $x_{i,j}$ and $x_{j,i}$ are different variables). Hence, X is the set of all variables $x_{i,j}$.

A nonempty set of vertex disjoint cycles in \vec{G} is a *cycle packing*. It corresponds to a set of cycles in G and edges of G (which have become directed cycles of length 2). Let \mathcal{C} be a minimum weight cycle packing. If no edge has weight 0, \mathcal{C} is either an orientation of a shortest cycle of G , or a directed cycle of length 2. If there are edges of weight 0, \mathcal{C} is a set of directed cycles of weight 0, possibly including a shortest cycle of G . The following lemma checks if there is a cycle packing of weight d that is not composed solely of cycles of length 2.

Lemma IV.1. *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: By definition

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

where Γ_n is the set of n -element permutations, and $\sigma(p)$ is the sign of the permutation p . p defines a set of directed edges $\mathcal{C}_p = \{(i, p_i) : 1 \leq i \leq n\}$. The cycles of \mathcal{C}_p are the cycles of p . \mathcal{C}_p includes self-loops for all i such that $i = p_i$. p corresponds to a non-zero term in the determinant iff \mathcal{C}_p contains only oriented edges from E or self-loops. Hence, after throwing away self-loops p can be identified with a cycle packing in G , or \emptyset . The term corresponding to a set of self-loops is equal to 1. Hence, $\det(\tilde{A}(G) + I) - 1$ contains only terms that correspond to cycle packings. Moreover the degree of y in the term for p is the total weight of the cycles in \mathcal{C}_p , where self-loops have weight zero.

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

Algorithm 1 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 $\delta'_{u,v} = \left[x_{u,v} \delta_{u,v} - x_{v,u} \delta_{v,u} \right] |_\sigma$.
 - 5: Return true if δ' has a non-zero entry.
-

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} . ■

Call an edge *allowed* iff it belongs to some shortest cycle. The above proof characterizes these edges: For any $d \in [0, nW]$ define the set $\{uv \in E : \left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] \text{term}_y^d \left[\det(\tilde{A}(G) + I) - 1 \right] \neq 0\}$. Taking d as the smallest value where this set is nonempty gives the set of allowed edges.

We cannot find this smallest d efficiently. But we can efficiently test if it is $\leq c$ for any c we choose. The idea is to use polynomial multiplication to combine all the terms of degree $\leq c$: Take any $c \in [0, nW]$. There exists $d \leq c$ such that $\left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] \text{term}_y^d \left[\det(\tilde{A}(G) + I) - 1 \right] \neq 0$ iff $\left[x_{u,v} \frac{\partial}{\partial x_{u,v}} - x_{v,u} \frac{\partial}{\partial x_{v,u}} \right] \text{term}_y^c \left[\left(\det(\tilde{A}(G) + I) - 1 \right) \cdot \left(\sum_{i=0}^{nW} y^i \right) \right] \neq 0$. (It is easy to see that multiplying by $\sum_{i=0}^{nW} y^i$ causes no cancellation.) This leads to Algorithm 1.

Using binary search with this algorithm we obtain:

Theorem IV.2. *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.*

It is a simple matter to find a shortest cycle, given its weight c^* found as above: Let δ' be the matrix computed by Algorithm 1 for $c = c^*$. Take any edge uv such that $\delta'_{uv} \neq 0$. Use Dijkstra's algorithm to find a shortest path

$p_{v,u}$ from v to u in $G \setminus \{uv\}$. The cycle formed by uv and $p_{v,u}$ is shortest.

V. 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 2). 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 3). 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 4). 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 V.11). 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 V-B 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 V.19).

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

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 V-A). Next, in Section V-B 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 V-C, we gather all the theorems and formally prove correctness and bound the running time of our algorithm.

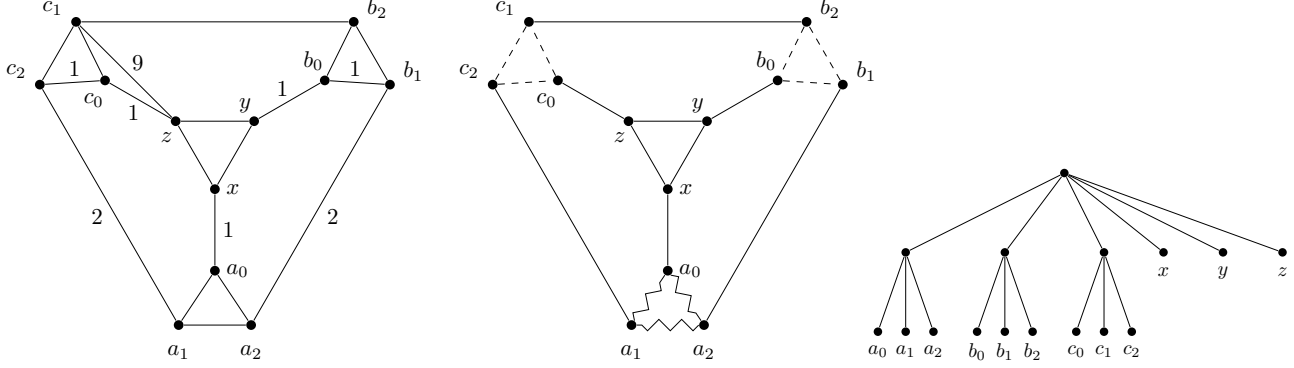


Figure 1. The matching algorithm: The far left shows an example graph. Edges without a label weigh 0. The minimum weight of a perfect matching is 3. $w(M(v))$ is 2 for $v \in \{b_2, c_1, x, y, z\}$ and 1 otherwise. The middle figure shows the allowed edges (edge c_1z was removed) and new edge weights $w'(uv) = w(uv) + w(M(u)) + w(M(v))$; 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 4.

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 V-B and in the algorithm for finding unweighted maximum matching problem.

A. 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 [16] 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 V.1 (♠). *An edge $ij \in E$ belongs to some minimum weight perfect matching iff*

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

Corollary V.2. *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.*

Definition V.3 ($M(uv)$, $M(u)$). 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,

Algorithm 2 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's 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 $\delta_{i,j} \neq 0$.

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 V.4 (♠). *Let $G = (V, E, w, W)$ be a weighted undirected graph. Then $\deg_y^* (\text{adj}(\tilde{A}(G))_{i,j}) = w(M) + w(M(ij))$.*

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 V.5. *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 V.4 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 $\deg_y^*(\text{adj}(\tilde{A}(G))b) = w(M) + w(M(i))$. ■

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

- 1: Generate a random substitution $\sigma : X \cup B \rightarrow \mathbb{Z}_p$ 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 V.5 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 V.6. Algorithm 3 computes the values $w(M(u))$, for all $u \in V(G)$, in $\tilde{O}(Wn^\omega)$ time, with high probability.

B. Properties of the Dual

We move on to the linear programming formulation of the minimum weight perfect matching problem given by Edmonds [17]. An *odd set* has odd cardinality; Ω denotes the collection of odd subsets of V of cardinality ≥ 3 .

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

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 :

$$\begin{aligned} \max \sum_{v \in V} \pi_v + \sum_{U \in \Omega} \pi_U \\ \pi_u + \pi_v + \sum_{\substack{U \in \Omega \\ uv \in \delta(U)}} \pi_U &\leq w(uv) \text{ for all } uv \in E \quad (2) \\ \pi_U &\geq 0 \text{ for all } U \in \Omega \end{aligned}$$

We say that an edge $e = uv$ is *tight* with respect to a dual π if equality holds in (2). 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 V.7 (Edmonds '65 [17]). *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 V.8 (critical dual, blossom). An optimum dual solution satisfying the conditions from Lemma V.7 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 V.9. *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 [18]–[20] 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 $\tilde{O}(|U|^\omega)$. ■

Complementary slackness gives the following observation, which together with Lemma V.9 suffice to prove Lemma V.11.

Observation V.10. *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 V.11 (♠). *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.*

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

Definition V.12 (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 V.13. *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 V.10(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. ■

The next steps of our development (Lemmas V.16–V.18) can be derived using an appropriate version of Edmonds’ weighted matching algorithm (e.g., [21]). Here we will use a structural approach, based on the following properties of allowed edges given by Lovász and Plummer.

Lemma V.14 ([22], 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 V.15 (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 V.16. *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 $w(M(G)) - H(T(\pi)) - dist_{T(\pi)}(z, V)$.*

Proof: Any almost perfect matching in $G \setminus z$ weighs at least $w(M(G)) - H(T(\pi)) - dist_{T(\pi)}(z, V)$. 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 (2) 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 $w(M(G)) - dist_{T(\pi)}(z, V) - dist_{T(\pi)}(x, V)$. The definition of $H(T(\pi))$ shows the last quantity is at least $w(M(G)) - dist_{T(\pi)}(z, V) - H(T(\pi))$ as desired.

We complete the proof by constructing an almost perfect matching in $G \setminus z$ of weight $w(M(G)) - H(T(\pi)) - dist_{T(\pi)}(z, V)$. Take G', u, v, u', v' as in Definition V.15. 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 V.14. Definition V.15 shows that $u' \not R v'$. So z' is not equivalent to at least of u and v . W.l.o.g. assume that $u' \not R z'$. 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 V.9.

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. ■

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 V.17. *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 V.14 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 V.14 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.

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 2, so it remains tight. Next observe that the blossoms of π_1 form a laminar family. Lemma V.14 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.

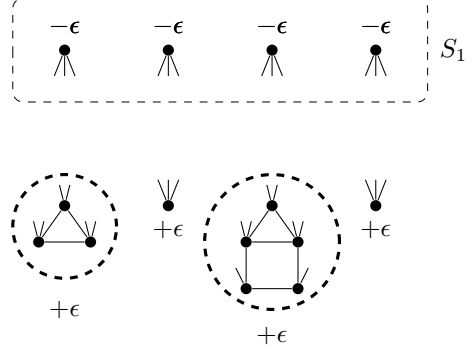


Figure 2. Graph G' and the modification of the duals.

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 V.14 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 . ■

Lemma V.18. *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 V.7 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 (2) 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 (2) 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} . ■

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 V.19. *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 V.16 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

$$\begin{aligned} w'(uv) &= w(uv) + w(M(u)) + w(M(v)) \\ &= \text{dist}_{T(\pi)}(u, v) + 2w(M(G)) - 2H(T(\pi)) \\ &\quad - \text{dist}_{T(\pi)}(u, V) - \text{dist}_{T(\pi)}(v, V) \\ &= c - 2\text{dist}_{T(\pi)}(\text{lca}(u, v), V). \end{aligned} \quad (3)$$

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

Algorithm 4 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} .
-

let $B_{uv} \subseteq V$ be the 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 (3). 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, (3) 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 $B_{uv} \subseteq B$. ◊

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 4. ■

C. The Final Algorithm

Theorem V.20. *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 V.2, 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 V.6. Having all the values $w(M(u))$ by Lemma V.19 we can find the set of blossoms \mathcal{B} of a balanced critical dual solution and consequently by Lemma V.11 we can find a minimum weight perfect matching in G . ■

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 and its weight is easily found. As discussed in the proof of Lemma V.4, 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. Consequently 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 plus the weight of a second smallest perfect matching.

VI. FURTHER APPLICATIONS

Several other ideas are used in the applications covered in the full version [1]. As in Corollary V.5, terms of the adjoint matrix are often useful, but it is difficult to compute the entire adjoint. Sometimes we use the $n \times n$ symbolic matrix \tilde{Z} given by $\tilde{Z}_{i,j} = z_{i,j}$, where the $z_{i,j}$ are n^2 new indeterminates. For any $n \times n$ matrix \tilde{A} not involving any $z_{i,j}$, and σ_z the evaluation that assigns 0 to each $z_{i,j}$,

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

For shortest path problems on undirected graphs with negative weights we use a construction which we believe is essentially due to Edmonds [5]: Letting E^- be the set of negative edges, define the graph \tilde{G} by

$$\begin{aligned} \tilde{V} &= \{v_1, v_2 : v \in V\} \cup \{e_1, e_2 : e \in E^-\}, \\ \tilde{E} &= \{u_1v_2, u_2v_1, u_1v_1, u_2v_2 : uv \in E \setminus E^-\} \\ &\cup \{u_1e_1, u_2e_1, e_1e_2, v_1e_2, v_2e_2 : e = uv \in E^-, u < v\} \\ &\cup \{v_1v_2 : v \in V\} \\ \ddot{w}(u_iv_j) &= \begin{cases} w(uv) & \text{if } uv \in E \setminus E^-, \\ w(e) & \text{if } u_i = e_1 \text{ and } v_j \neq e_2 \text{ and } e \in E^-, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

\tilde{G} has $O(n)$ vertices and $O(n+m)$ edges (since $|E^-| < n$ when G has no negative cycles). A shortest path from u to v in G corresponds to a minimum weight perfect matching on $\tilde{G} \setminus \{u_2, v_2\}$. For diameter and radius we use \tilde{Z} on \tilde{G} . For shortest cycles we eliminate spurious cycles like $(u_1, v_1), (v_2, u_2)$ using antisymmetric derivatives (Section IV).

REFERENCES

- [1] M. Cygan, H. N. Gabow, and P. Sankowski, "Algorithmic applications of baur-strassen's theorem: Shortest cycles, diameter and matchings," *CoRR*, vol. abs/1204.1616, 2012.
- [2] A. Itai and M. Rodeh, "Finding a minimum circuit in a graph," in *Proc. of STOC'77*, 1977, pp. 1–10.
- [3] L. Roditty and V. V. Williams, "Minimum weight cycles and triangles: Equivalences and algorithms," in *Proc. of FOCS'11*, 2011, pp. 180–189.
- [4] P. Sankowski, "Shortest paths in matrix multiplication time," in *Proc. of ESA'05*, 2005, pp. 770–778.
- [5] J. Edmonds, "An introduction to matching. Mimeographed notes, Engineering Summer Conference, U. Michigan, Ann Arbor, MI," 1967.
- [6] A. Shoshan and U. Zwick, "All pairs shortest paths in undirected graphs with integer weights," in *Proc. of FOCS'99*, 1999, pp. 605–614.
- [7] P. Sankowski, "Maximum weight bipartite matching in matrix multiplication time," *Theoretical Computer Science*, vol. 410, no. 44, pp. 4480–4488, 2009.
- [8] C.-C. Huang and T. Kavitha, "Efficient algorithms for maximum weight matchings in general graphs with small edge weights," in *Proc. of SODA'12*, 2012, pp. 1400–1412.
- [9] A. Storjohann, "High-order lifting and integrality certification," *Journal of Symbolic Computation*, vol. 36, no. 3-4, pp. 613–648, 2003.
- [10] W. Baur and V. Strassen, "The complexity of partial derivatives," *Theoretical Computer Science*, vol. 22, no. 3, pp. 317–330, 1983.
- [11] J. Morgenstern, "How to compute fast a function and all its derivatives: a variation on the theorem of Baur-strassen," *SIGACT News*, vol. 16, no. 4, pp. 60–62, 1985.
- [12] R. Yuster, "A shortest cycle for each vertex of a graph," *Information Processing Letters*, vol. 111, no. 21-22, pp. 1057–1061, 2011.
- [13] R. Zippel, "Probabilistic algorithms for sparse polynomials," in *Proc. of EUROSAM'79*, 1979, pp. 216–226.
- [14] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM*, vol. 27, pp. 701–717, 1980.
- [15] A. Björklund, "Determinant sums for undirected hamiltonicity," in *Prof. of FOCS'10*, 2010, pp. 173–182.
- [16] R. M. Karp, E. Upfal, and A. Wigderson, "Constructing a perfect matching is in random NC," *Combinatorica*, vol. 6, no. 1, pp. 35–48, 1986.
- [17] J. Edmonds, "Maximum matching and a polyhedron with 0,1-vertices," *Journal of Research National Bureau of Standards-B.*, vol. 69B, pp. 125–130, 1965.
- [18] M. Mucha and P. Sankowski, "Maximum matchings via Gaussian elimination," in *Proc. of FOCS'04*, 2004, pp. 248–255.
- [19] P. Sankowski, "Processor efficient parallel matching," in *Proc. of SPAA'05*, 2005, pp. 165–170.
- [20] N. J. A. Harvey, "Algebraic structures and algorithms for matching and matroid problems," in *Proc. of FOCS'06*, 2006, pp. 531–542.
- [21] A. Schrijver, *Combinatorial Optimization - Polyhedra and Efficiency*. Springer-Verlag, 2003.
- [22] L. Lovász and M. D. Plummer, *Matching Theory*. Akadémiai Kiadó, 1986.