

On Finding ℓ -Th Smallest Perfect Matchings

Nicolas El Maalouly 

Department of Computer Science, ETH Zurich, Switzerland

Sebastian Haslebacher 

Department of Computer Science, ETH Zurich, Switzerland

Adrian Taubner 

Department of Computer Science, ETH Zurich, Switzerland

Lasse Wulf 

Section of Algorithms, Logic, and Graphs, Technical University of Denmark, Lyngby, Denmark
IT University, Copenhagen, Denmark

Abstract

Given an undirected weighted graph G and an integer k , Exact-Weight Perfect Matching (EWPM) is the problem of finding a perfect matching of weight exactly k in G . In this paper, we study EWPM and its variants. The EWPM problem is famous, since in the case of unary encoded weights, Mulmuley, Vazirani, and Vazirani showed almost 40 years ago that the problem can be solved in randomized polynomial time. However, up to this date no derandomization is known.

Our first result is a simple deterministic algorithm for EWPM that runs in time $n^{\mathcal{O}(\ell)}$, where ℓ is the number of distinct weights that perfect matchings in G can take. In fact, we show how to find an ℓ -th smallest perfect matching in any weighted graph (even if the weights are encoded in binary, in which case EWPM in general is known to be NP-complete) in time $n^{\mathcal{O}(\ell)}$ for any integer ℓ . Similar next-to-optimal variants have also been studied recently for the shortest path problem.

For our second result, we extend the list of problems that are known to be equivalent to EWPM. We show that EWPM is equivalent under a weight-preserving reduction to the Exact Cycle Sum problem (ECS) in undirected graphs with a conservative (i.e. no negative cycles) weight function. To the best of our knowledge, we are the first to study this problem. As a consequence, the latter problem is contained in RP if the weights are encoded in unary. Finally, we identify a special case of EWPM, called BCPM, which was recently studied by El Maalouly, Steiner and Wulf. We show that BCPM is equivalent under a weight-preserving transformation to another problem recently studied by Schlotter and Sebő as well as Geelen and Kapadia: the Shortest Odd Cycle problem (SOC) in undirected graphs with conservative weights. Finally, our $n^{\mathcal{O}(\ell)}$ algorithm works in this setting as well, identifying a tractable special case of SOC, BCPM, and ECS.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Problems, reductions and completeness; Theory of computation $\rightarrow$ Randomness, geometry and discrete structures; Theory of computation $\rightarrow$ Design and analysis of algorithms

Keywords and phrases Exact Matching, Perfect Matching, Exact-Weight Perfect Matching, Shortest Odd Cycle, Exact Cycle Sum, ℓ -th Smallest Solution, ℓ -th Largest Solution, k -th Best Solution, Derandomization

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.19

Related Version *Full Version*: <https://doi.org/10.48550/arXiv.2506.22619>

Funding *Lasse Wulf*: Supported by Carlsberg Foundation CF21-0302 “Graph Algorithms with Geometric Applications”

1 Introduction

A big open question in complexity theory concerns the role of randomness in devising polynomial-time algorithms. While it is conjectured that every problem in RP (and even BPP) should also lie in P (see e.g. Chapter 20 in [1] for more details), there exist several



© Nicolas El Maalouly, Sebastian Haslebacher, Adrian Taubner, and Lasse Wulf;
licensed under Creative Commons License CC-BY 4.0

33rd Annual European Symposium on Algorithms (ESA 2025).

Editors: Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman; Article No. 19; pp. 19:1–19:15

Leibniz International Proceedings in Informatics



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

natural problems for which efficient randomized algorithms, but no efficient deterministic algorithms, are known. Thus, instead of attacking the conjecture $\text{RP} = \text{P}$ directly, it might be more reasonable to first try to devise efficient deterministic algorithms for those problems.

A famous example of a problem in RP , that is not known to lie in P , is the unary version of Exact-Weight Perfect Matching (EWPM): Given an undirected weighted graph G and an integer k , EWPM is the problem of deciding whether G admits a perfect matching of weight exactly k . Note that this problem is often called Exact Matching (EM) if we allow only weights 0 and 1, and it is not hard to see (by replacing single edges by paths corresponding to their weight) that unary EWPM is equivalent to EM.

EWPM was introduced by Papadimitriou and Yannakakis [27] in 1982, and it is known to be NP -hard in general (for a short proof, see [14]). Papadimitriou and Yannakakis also conjectured that EWPM remains NP -hard if weights are encoded in unary. However, Mulmuley, Vazirani and Vazirani [24] proved in 1987 that unary EWPM is in RP . This makes it very unlikely for the problem to be NP -hard. Hence, the question became whether one could also find an efficient deterministic algorithm for unary EWPM.

This question has been open for almost 40 years now. In fact, despite considerable effort, no deterministic algorithm is known that runs in time that is subexponential in the number of vertices of the graph. Instead, progress has been made mostly by focusing on restricted graph classes, approximation variants of EWPM, and finding other natural problems that are equivalent to unary EWPM (see Section 1.3 for more details).

Our main result is a new parameterized algorithm for EWPM that works on general graphs, even if weights are encoded in binary: Assume that $k_1 < k_2 < \dots < k_\ell$ are all the distinct weights up to k_ℓ that perfect matchings in G take. We prove that it is possible to find a perfect matching of weight k_ℓ in time $n^{\mathcal{O}(\ell)}$. Our algorithm is very simple and relies on our main theorem (Theorem 9), which says that there must exist $2(\ell - 1)$ edges such that any minimum-weight perfect matching that includes all of these $2(\ell - 1)$ edges must have weight k_ℓ . The obtained bound of $2(\ell - 1)$ is tight.

1.1 Next-to-Optimal Problems

Our main result can be equivalently stated in a different way: Consider the following problem, which we call ℓ -th Smallest Perfect Matching (SPM_ℓ): Given a weighted graph G , an integer k , and a natural number ℓ , decide whether G contains an ℓ -th smallest perfect matching of weight exactly k . Here, the notion of ℓ -th smallest perfect matchings is defined recursively: 1st-smallest perfect matchings are exactly the minimum-weight perfect matchings, and for $\ell \geq 2$, ℓ -th smallest perfect matchings are the perfect matchings of minimum weight among all perfect matchings that are not ℓ' -th smallest for any $\ell' < \ell$. In this language, our result is an algorithm that solves SPM_ℓ in time $n^{\mathcal{O}(\ell)}$.

Similar next-to-optimal problems have been studied in the literature before. In particular, there is a line of work on finding next-to-shortest paths in directed or undirected graphs. In these problems, we want to find a next-to-shortest (s, t) -path in a weighted graph G , i.e. the path should have minimum weight among all (s, t) -paths that are not shortest paths. This problem is NP -hard on directed graphs [22], but admits polynomial-time algorithms on undirected graphs [18, 21, 23, 31, 33]. Note that the NP -hardness result in [22] heavily relies on allowing edges to have weight zero. Indeed, the so-called detour problem – decide whether all (s, t) -shortest paths in a given unweighted directed graph have the same length or not – is very similar but does not allow for zero-weight edges, and its complexity status is still unresolved [15].

Instead of asking for a perfect matching of ℓ -th smallest weight, one could instead ask for ℓ distinct perfect matchings while minimizing their combined weight. This is a conceptually different problem which is equivalent to repeatedly (ℓ times) finding the best solution distinct from previously obtained solutions. Such k -best enumeration problems have been studied for a number of combinatorial optimization problems: Murty [26] first showed a polynomial-time algorithm for the k -best bipartite perfect matching problem. Later, the time complexity was improved by Chegireddy and Hamacher [2]. For more information, we refer the reader to the survey by Eppstein [11].

1.2 Related Cycle Problems

Our second contribution in this paper is to strengthen the connection between EWPM and related cycle problems, as we will outline here.

Consider the following problem, which we call Exact Cycle Sum (ECS): Given an undirected graph G with conservative (i.e. no negative cycles) edge-weights and an integer k , decide whether G contains a set of disjoint cycles of weight exactly k . A directed version of ECS with general weights has been studied e.g. by Papadimitriou and Yannakakis [27]. Concretely, they proved that the directed version of ECS is equivalent to EWPM on bipartite graphs. We extend this result by proving that the undirected problem ECS as defined above is equivalent to EWPM on general graphs. Our reductions preserve the encoding of the weights. Hence, we obtain as a result that ECS is in RP if weights are encoded in unary, and NP-complete if weights are encoded in binary.

Another interesting problem on cycles in graphs is the so-called Shortest Odd Cycle problem (SOC): Given an undirected graph G with conservative edge-weights and an integer k , decide whether G contains a cycle of odd weight at most k . SOC was recently studied by Schlotter and Sebő [28] (they use slightly different but equivalent definition, see Section 2.3 for more details), who proved that it is equivalent to finding minimum-weight odd T -joins. They also point out that SOC is contained in RP for unary weights, due to a result by Geelen and Kapadia [13], and therefore conjecture that the problem should be contained in P as well. Interestingly, the algorithm in [13] follows from a reduction to a problem on perfect matchings that can then be solved using the techniques of Mulmuley, Vazirani, and Vazirani [24]. In fact, this problem on perfect matchings has independently been studied by El Maalouly, Steiner, and Wulf [8] who called it Bounded Correct-Parity Perfect Matching (BCPM): Given a weighted graph G and an integer k , decide whether G contains a perfect matching M of weight $w(M)$ at most k and with $w(M) \equiv_2 k$ (where $\equiv_2$ is used to denote mod 2 equivalence). El Maalouly, Steiner, and Wulf use BCPM on bipartite graphs as a subroutine for a parameterized algorithm for EM on bipartite graphs. To this end, they prove that BCPM on bipartite graphs is in P, by reducing it to the directed version of SOC (which can be solved via dynamic programming). In this paper, we extend this set of results by proving that BCPM on general graphs is equivalent to SOC as defined above. Unfortunately, we are not able to give deterministic polynomial-time algorithms for either problem, but any such deterministic polynomial-time algorithm for SOC or BCPM would immediately imply that the parameterized algorithm from [8] also works on general graphs. Furthermore, our results highlight that the two equivalent problems SOC and BCPM can be seen as a stepping stone for solving unary EWPM, and that it might be reasonable to tackle them first.

Concretely, some interesting further questions that remain unresolved are: Can we solve SOC and BCPM in deterministic polynomial time? The conjecture $\text{RP} = \text{P}$ suggests that for unary weights, one should be able to do it. Interestingly, we know very little about the case of weights encoded in binary: If the weights are encoded in binary, it is not known whether SOC or BCPM are in RP or NP-hard (since the randomized algorithm in [13] works only for unary weights). Note that the same questions were also asked by Schlotter and Sebő [28].

While in the general case, derandomization of unary SOC and BCPM remains open, combining our algorithm for EWPM with our reductions also implies that BCPM, SOC, and ECS can be deterministically solved in time $n^{O(\ell)}$ if at most ℓ many different possible weights of perfect matchings (cycle sums, respectively) appear in the instance.

1.3 Further Related Work on EM

As mentioned before, EWPM with weights restricted to 0 and 1 is called the Exact Matching Problem (EM). Derandomizing the known randomized algorithm is an important open problem. Solution approaches have mainly been focused on three directions: Finding deterministic approximation algorithms, finding deterministic algorithms on restricted graph classes, and identifying other polynomial-time equivalent problems.

- In terms of approximation algorithms for EM, Yuster [32] proved that one can find an almost exact matching (a matching of weight exactly k that may fail to cover two vertices of the graph) in deterministic polynomial time if an exact matching exists in the graph. One could interpret this as an approximation result that relaxes the perfect matching constraint. If we relax the exactness constraint instead, recent progress includes a two-sided 2-approximation algorithm on bipartite graphs by El Maalouly [5], and a one-sided 3-approximation algorithm on bipartite graphs by Dürr, El Maalouly, and Wulf [3].
- EM has been studied on various restricted graph classes. In fact, Karzanov [19] proved already in 1987 that EM can be solved in deterministic polynomial time on complete and complete bipartite graphs. This has been generalized in different ways: El Maalouly and Steiner [7] gave deterministic polynomial-time algorithms on graphs with bounded independence number (and bipartite graphs of bounded bipartite independence number). In the case of bipartite graphs, this was improved to an FPT-algorithm by El Maalouly, Steiner, and Wulf [8]. In the general case, Murakami and Yamaguchi [25] gave an FPT-algorithm that is parametrized by the independence number and the minimum size of an odd cycle transversal.

El Maalouly, Haslebach, and Wulf [6] used Karzanov's original techniques to prove that EM restricted to chain graphs, unit-interval graphs, and complete r -partite graphs is in P. They also showed that a local search algorithm works in deterministic polynomial time on complete r -partite graphs, graphs of bounded neighborhood diversity, and graphs without complete bipartite t -holes.

Finally, it is also possible to derandomize the randomized algorithm on some graph classes, e.g. by finding so-called Pfaffian orientations. Concretely, Vazirani [30] showed how to do this for $K_{3,3}$ -free graphs, and Galluccio and Loeb [12] obtained Pfaffian orientations for graphs of bounded genus.

- Previously, some effort has also been made to identify other natural problems that are polynomial-time equivalent to EM. We already mentioned some of these results in Section 1.2. Besides the mentioned cycle problems, EM is also known to be equivalent to a problem called Top-k Perfect Matching [9].
- Finally, Jia, Svensson, and Yuan [16] recently proved that the bipartite exact matching polytope has exponential extension complexity. This stands in contrast to the bipartite perfect matching polytope, which is known to be integral.

We remark that many of these algorithmic results only hold for 0-1-weighted EWPM, and do not even translate to unary EWPM since they often involve very specific graph structures or parameters. In contrast, our parameterized algorithm works even for binary encoding of weights in EWPM.

1.4 Outline

We formally define EWPM and related problems in Section 2, where we also recall some of the most important concepts from the literature. In Section 3, we introduce the ℓ -th Smallest Perfect Matching (SPM_ℓ) problem, and we prove that it can be solved deterministically in time $n^{\mathcal{O}(\ell)}$. In Section 4, we explain our reductions between cycle and matching problems. Full proofs of all statements in Section 4 can be found in the Appendix of the full version.

2 Preliminaries

We start by introducing notation and afterwards recall important facts from the literature. First of all, all our graphs are considered to be simple, i.e. they do not contain self-loops or parallel edges. If not specified, graphs are also considered to be undirected. Now, given a graph $G = (V, E)$, we will frequently use the standard notions of paths, cycles, and matchings, and we usually think of those objects as subsets of edges. In particular, a perfect matching M of G is a subset of edges $M \subseteq E$ such that every vertex of G is covered exactly once by M . We denote by $\mathcal{M}_G$ the set of all perfect matchings of G .

Occasionally, we will explicitly use directed graphs. In particular, we will write $(u, v) \in E$ (instead of $\{u, v\} \in E$) for directed edges in a directed graph $G = (V, E)$ with $u, v \in V$.

In some problems, edge-weights are given for a graph $G = (V, E)$ in terms of a function $w : E \rightarrow \mathbb{Z}$ or $w : E \rightarrow \mathbb{N}$. We use the convention that $w(X)$ denotes the sum of all weights of edges in $X \subseteq E$.

For two decision problems A and B , we use $A \leq_p B$ to denote that there exists a polynomial-time many-one reduction from A to B . We also write $A \equiv_p B$ to say that both $A \leq_p B$ and $B \leq_p A$ are true.

2.1 Symmetric Difference and Alternating Cycles

An important and common tool for working with perfect matchings is the *symmetric difference*. The symmetric difference of two sets X, Y is defined as $X \Delta Y := (X \cup Y) \setminus (X \cap Y)$. It is a well-known fact that the symmetric difference $M \Delta M'$ of two perfect matchings M, M' in a graph $G = (V, E)$ always consists of a set of vertex-disjoint even cycles. In other words, there exist vertex-disjoint even cycles $C_1, \dots, C_p \subseteq E$ such that $M \Delta M' = \bigsqcup_{i=1}^p C_i$. We say that C_i is a cycle from $M \Delta M'$, and we sometimes use the letter p to refer to the number of cycles in $M \Delta M'$. Further, we recall that every cycle C in $M \Delta M'$ is alternating in both M and M' (an even cycle C of G is called *M -alternating* if and only if it alternates between edges in M and edges not in M). The set $M \Delta C$ is again a matching of the same cardinality as M . We often refer to this property by saying that we *switch* M along C to obtain the new matching $M \Delta C$. Naturally, the matchings $M \Delta C$ and M differ exactly in C , i.e. $(M \Delta C) \Delta M = C$.

Recall again that $M \Delta M'$ is a collection of vertex-disjoint even cycles $C_1, \dots, C_p \subseteq E$, and that each of the cycles $C_1, \dots, C_p$ is both M -alternating and M' -alternating. Hence, given both M, M' , some new perfect matchings of G can be obtained by switching either M or M' along a subset of the cycles $C_1, \dots, C_p$. In particular, we can perform the switching operations independently of each other because the cycles are vertex-disjoint.

2.2 Minimum-Weight and Exact-Weight Perfect Matching

Finding a minimum-weight perfect matching is a classical problem that has been studied extensively. We will define it as follows.

► **Definition 1** (MWPM). *Given a graph $G = (V, E)$, edge-weights $w : E \rightarrow \mathbb{Z}$, and an integer k , Minimum-Weight Perfect Matching (MWPM) is the problem of deciding whether G admits a perfect matching of total weight at most k .*

It is well-known that MWPM is in P (even if weights are encoded in binary, see e.g. [20]). Moreover, finding a maximum-weight perfect matching is analogous to finding a minimum-weight perfect matching, and hence that can also be done in deterministic polynomial time. Since both of these problems are in P, it then seems natural to ask whether we can also test for a perfect matching of weight exactly k in polynomial time.

► **Definition 2** (EWPM). *Given a graph $G = (V, E)$, edge-weights $w : E \rightarrow \mathbb{Z}$, and an integer k , Exact-Weight Perfect Matching (EWPM) is the problem of deciding whether G admits a perfect matching of weight exactly k .*

As mentioned in Section 1, the variant of EWPM with weights from $\{0, 1\}$ is also known as Exact Matching (EM), and it is contained in RP [24]. Moreover, this also implies that EWPM with weights encoded in unary is contained in RP. In contrast, EWPM with weights encoded in binary is NP-complete [27, 14]. Note that this constitutes a major difference to MWPM, where both the unary and binary variant admit efficient algorithms.

The following problem BCPM is at most as hard as EWPM, and it was used in [8] as part of an algorithm for EM.

► **Definition 3** (BCPM). *Given a graph $G = (V, E)$, edge-weights $w : E \rightarrow \mathbb{Z}$, and an integer k , Bounded Correct-Parity Perfect Matching (BCPM) is the problem of deciding whether G admits a perfect matching M of weight $w(M)$ at most k satisfying $k \equiv_2 w(M)$.*

The randomized algorithm for unary EWPM also implies a randomized algorithm for unary BCPM. Interestingly, the complexity of binary BCPM remains open. In particular, it is not known whether binary BCPM is NP-hard or not.

2.3 Conservative Weights and Cycle Problems

Many problems on finding shortest paths and cycles in graphs with non-negative edge-weights can be solved efficiently. On the other hand, allowing for negative weights usually makes those problems NP-hard as they now include problems such as finding Hamiltonian paths or cycles. An interesting compromise is provided by conservative edge-weights.

► **Definition 4** (Conservative Weights). *Edge-weights $w : E \rightarrow \mathbb{Z}$ of a (possibly directed) graph $G = (V, E)$ are conservative if and only if there are no negative (directed) cycles in G with respect to w .*

Using established techniques (see e.g. Johnson's algorithm [17] or Suurballe's algorithm [29]), conservative weights on directed graphs can be turned into non-negative weights while maintaining the total weight of each cycle. Hence, cycle problems in directed graphs with conservative edge-weights are usually not harder than their non-negative variants. However, in the case of undirected graphs, the same technique is not applicable, and it turns out that there are interesting cycle problems on undirected graphs with conservative edge-weights that relate to EWPM and BCPM. Concretely, we will work with the two problems Exact Cycle Sum (ECS) and Shortest Odd Cycle (SOC), defined on graphs with conservative weights.

► **Definition 5** (ECS). *Given a graph $G = (V, E)$ with conservative edge-weights $w : E \rightarrow \mathbb{Z}$ and an integer k , Exact Cycle Sum (ECS) is the problem of deciding whether there exists a set of vertex-disjoint cycles of total weight exactly k in G .*

► **Definition 6 (SOC).** *Given a graph $G = (V, E)$ with conservative edge-weights $w : E \rightarrow \mathbb{Z}$ and an integer k , Shortest Odd Cycle (SOC) is the problem of deciding whether there exists a cycle of odd weight at most k in G .*

Note that the definition of SOC by Schlotter and Sebő [28] as well as Geelen and Kapadia [13] slightly differs from ours: They require the cycle to have odd length (i.e. an odd number of edges) instead of odd weight. However, it is not hard to see that the two versions are polynomial-time equivalent. Indeed, by splitting even-weight edges, we can ensure to get a graph with only odd-weight edges, yielding a reduction from our odd-weight version to the odd-length version. To reduce the odd-length version to the odd-weight version, it suffices to manipulate the weights, replacing the weight $w(e)$ of each edge e with $2|V|w(e) + 1$.

As mentioned in Section 1.2, El Maalouly et al. [8] gave a polynomial-time algorithm for BCPM on bipartite graphs by reducing it to a directed version of SOC. An analogous connection between EWPM on bipartite graphs and the directed version of ECS has been observed by Papadimitriou and Yannakakis [27] already in 1982. In Section 4, we will extend these connections between cycle and matching problems by proving equivalence of ECS (undirected) and EWPM (on general graphs), and SOC (undirected) and BCPM (on general graphs), respectively.

3 ℓ -th Smallest Perfect Matchings

As discussed in Section 1, the problem of finding a deterministic polynomial-time algorithm for unary EWPM has been attacked from various directions in recent years. We attack it from yet another direction that involves so-called ℓ -th smallest perfect matchings.

► **Definition 7.** *Let $G = (V, E)$ be a graph with edge-weights $w : E \rightarrow \mathbb{Z}$. We define ℓ -th smallest perfect matchings for all $\ell \in \mathbb{N}$ recursively. The 1st-smallest perfect matchings of G are exactly the minimum-weight perfect matchings. Moreover, for fixed $\ell \geq 2$, a perfect matching M is ℓ -th smallest if and only if it has minimum weight among all perfect matchings that are not ℓ' -th smallest for any $\ell' < \ell$.*

► **Definition 8 (SPM_ℓ).** *Given a graph $G = (V, E)$, edge-weights $w : E \rightarrow \mathbb{Z}$, an integer k , and a natural number ℓ , ℓ -th Smallest Perfect Matching (SPM_ℓ) is the problem of deciding whether G admits an ℓ -th smallest perfect matching of weight exactly k .*

The goal of this section is to give a deterministic algorithm that can decide SPM_ℓ in time $n^{\mathcal{O}(\ell)}$, where n is the size of the instance. Naturally, this has interesting consequences for EWPM: Assume that we are given an instance of EWPM where k is such that any perfect matching M with $w(M) = k$ is an ℓ' -th smallest perfect matching. Then we can verify existence of such a matching in time $n^{\mathcal{O}(\ell')}$ by simply calling the SPM_ℓ -algorithm repeatedly for $\ell = 1, \dots, \ell'$.

To the best of our knowledge, we are the first to observe a result of this kind. Our algorithm is very simple and follows a natural approach: For a given ℓ and for every set $F \subseteq E$ of size at most $2(\ell - 1)$, compute a minimum-weight perfect matching that includes all of F . We prove in Theorem 9 that there is always a 1st-smallest, a 2nd-smallest, $\dots$, an ℓ -th smallest perfect matching among the set of perfect matchings that were computed. The proof of Theorem 9 is less straight-forward. It relies on the complementary slackness theorem (compare e.g. [20]) and the perfect matching polytope.

► **Theorem 9** (ℓ -th Smallest PM in General Graphs). *Let $G = (V, E)$ be a graph with edge-weights $w : E \rightarrow \mathbb{Z}$. Let $\ell \in \mathbb{N}$ be arbitrary. Assume that G contains an ℓ -th smallest perfect matching M . Then there exists a subset of edges $F \subseteq E$ of size at most $2(\ell - 1)$ such that*

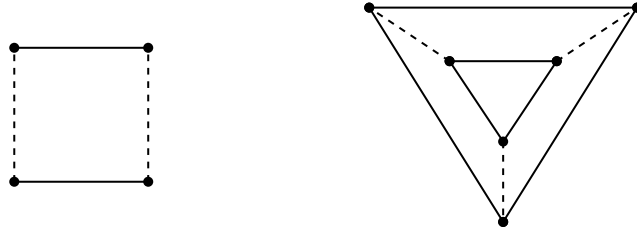
$$w(M) = \min_{M' \in \mathcal{M}_G : F \subseteq M'} w(M').$$

It turns out that one can slightly improve the above theorem in the case of bipartite graphs.

► **Theorem 10** (ℓ -th Smallest PM in Bipartite Graphs). *The ℓ -th smallest perfect matching in a weighted bipartite graph can be obtained the same way as in Theorem 9, except that F ranges only over edge sets of size at most $\ell - 1$.*

In this paper, we omit the proof of Theorem 10, since its proof is identical to that of Theorem 9, except that the general matching polytope is swapped with the bipartite matching polytope.

One might ask whether the number of edges one needs to “fix” in these two theorems is tight. Indeed, this is the case, i.e. in order to find the ℓ -th smallest perfect matching, one needs to fix $\ell - 1$ edges in a bipartite graph and $2(\ell - 1)$ edges in a general graph. To see this, consider Figure 1, where dashed edges have a weight of 1 while solid edges have a weight of 0. Consider first the bipartite four-cycle on the left. It has exactly two perfect matchings, one of weight 0 and one of weight 2. Fixing one edge is necessary and sufficient to find the 2nd-smallest perfect matching. Taking multiple disjoint copies of this four-cycle generalizes this to show that it is sometimes necessary to fix $\ell - 1$ edges to find the ℓ -th smallest perfect matching in a bipartite graph. Analogously, the graph on the right has perfect matchings of weight 1 and 3, but fixing a single edge does not suffice to find the perfect matching of weight 3. In other words, we need to fix 2 edges to find the second-smallest perfect matching here. Again, generalizing this argument by taking multiple copies of this graph proves that the bound $2(\ell - 1)$ is best possible in general graphs.



■ **Figure 1** Example that Theorems 9 and 10 are tight in terms of the number of fixed edges. Dashed edges have a weight of 1 and solid edges have a weight of 0.

3.1 Proof of Theorem 9

The goal of this section is to prove Theorem 9. We start by recalling some useful facts about the perfect matching polytope from Chapter 11 of the standard textbook by Korte and Vygen [20]. For this, it will be convenient to use the notation $\delta(v)$ to denote the set of edges incident to a vertex $v \in V$. Similarly, for all $A \subseteq V$, we use $\delta(A)$ to denote the set of edges with one endpoint in A and the other in $V \setminus A$. Moreover, we denote by $\mathcal{A}$ the set of all odd-cardinality sets $A \subseteq V$, i.e. $\mathcal{A} := \{A \subseteq V : |A| \equiv_2 1\}$.

For the remainder of this section, let $G = (V, E)$ be a graph with edge-weights $w : E \rightarrow \mathbb{Z}$. The integer linear program

$$\begin{aligned} & \min \sum_{e \in E} x_e w(e) \\ \text{s.t. } & \sum_{e \in \delta(v)} x_e = 1 & \forall v \in V \\ & \text{and } x_e \in \{0, 1\} & \forall e \in E \end{aligned}$$

with variables $\{x_e \mid e \in E\}$ captures the problem of finding a minimum-weight perfect matching in G . Now consider the following LP relaxation

$$\begin{aligned} & \min \sum_{e \in E} x_e w(e) \\ \text{s.t. } & \sum_{e \in \delta(v)} x_e = 1 & \forall v \in V \\ & \sum_{e \in \delta(A)} x_e \geq 1 & \forall A \in \mathcal{A} \text{ with } |A| > 1 \\ & x_e \geq 0 & \forall e \in E \end{aligned}$$

where additional odd-set constraints were added. By the famous result of Edmonds [4], adding these additional odd-set constraints is necessary and sufficient to make the LP integral. In other words, the basic feasible solutions to the LP are exactly the perfect matchings of G . The dual LP is now given by

$$\begin{aligned} & \max \sum_{A \in \mathcal{A}} z_A \\ \text{s.t. } & \sum_{A \in \mathcal{A} : e \in \delta(A)} z_A \leq w(e) & \forall e \in E \\ & z_A \geq 0 & \forall A \in \mathcal{A} \text{ with } |A| > 1 \end{aligned}$$

where we introduced a variable z_A for all $A \in \mathcal{A}$. By strong duality, the optimal value of the primal and the dual coincide.

► **Definition 11** (Residual Graph). *Let $G = (V, E)$ be a graph with edge-weights $w : E \rightarrow \mathbb{Z}$. Let z be an optimal solution to the dual LP associated with G . The graph $G_z := (V, E_z)$ with*

$$e \in E_z \iff w(e) = \sum_{A \in \mathcal{A} : e \in \delta(A)} z_A$$

is called residual graph with respect to z .

Unsurprisingly, it turns out that all minimum-weight perfect matchings are contained in the residual graph with respect to a given optimal solution z . However, there might exist perfect matchings in the residual graph that are not minimum-weight perfect matchings of G . Hence, we need an extra condition to exactly characterize which perfect matchings in the residual graph are minimum-weight in G . Eppstein [10] shows such a result for bipartite graphs. We present here the natural generalization to all graphs. This essentially follows from the complementary slackness theorem [20].

19:10 On Finding ℓ -Th Smallest Perfect Matchings

► **Lemma 12.** *Let $G = (V, E)$ be a graph with edge-weights $w : E \rightarrow \mathbb{Z}$. Let z be an optimal solution to the dual LP associated with G , and let $G_z = (V, E_z)$ be its residual graph. A perfect matching M of G has minimum weight if and only if $M \subseteq E_z$ and $|M \cap \delta(A)| = 1$ holds for all $A \in \mathcal{A}$ with $z_A > 0$.*

Proof. For the sake of clarity, we prove both directions individually.

($\Rightarrow$) We provide an indirect proof. Assume first that there exists $f \in M$ such that

$$\sum_{A \in \mathcal{A} : f \in \delta(A)} z_A < w(f)$$

holds and hence $f \notin E_z$. Observe that, by feasibility of z , we have

$$\sum_{A \in \mathcal{A} : e \in \delta(A)} z_A \leq w(e)$$

for all $e \in E$. Moreover, we have $z_A \geq 0$ for all $A \in \mathcal{A}$ with $|A| > 1$. Thus, we get

$$w(M) - \sum_{A \in \mathcal{A}} z_A \geq \sum_{e \in M} \left(w(e) - \sum_{A \in \mathcal{A} : e \in \delta(A)} z_A \right) > 0$$

since every set $A \in \mathcal{A}$ with $|A| = 1$ is associated with exactly one edge in M , and every other set $A \in \mathcal{A}$ (with $|A| > 1$) satisfies $z_A \geq 0$. In other words, we have

$$w(M) > \sum_{A \in \mathcal{A}} z_A$$

which, by strong duality, implies that M does not have minimum weight. Thus, assume now that $M \subseteq E_z$, but there exists $A \in \mathcal{A}$ such that $z_A > 0$ and $|M \cap A| > 1$. Again, we calculate

$$w(M) - \sum_{A \in \mathcal{A}} z_A > \sum_{e \in M} \left(w(e) - \sum_{A \in \mathcal{A} : e \in \delta(A)} z_A \right) = 0$$

where the strict inequality now comes from the fact that we needed to duplicate the variable z_A in the first step. Again, by strong duality, we conclude that M does not have minimum weight.

($\Leftarrow$) Assume now that we have $M \subseteq E_z$ and $|M \cap A| = 1$ for all $A \in \mathcal{A}$ with $z_A > 0$. In this case, we get

$$w(M) - \sum_{A \in \mathcal{A}} z_A = \sum_{e \in M} \left(w(e) - \sum_{A \in \mathcal{A} : e \in \delta(A)} z_A \right) = 0$$

as every set $A \in \mathcal{A}$ with $z_A > 0$ is associated with exactly one edge in M . By strong duality, we can use the fact that z is optimal to conclude that M has minimum weight. ◀

Using Lemma 12, we can prove the following lemma, which captures the special case $\ell = 2$ of Theorem 9.

► **Lemma 13.** *Let $G = (V, E)$ be a graph with edge-weights $w : E \rightarrow \mathbb{Z}$. Assume that G contains a 2nd-smallest perfect matching M . Then there exist edges $e, f \in E$ such that*

$$w(M) = \min_{M' \in \mathcal{M}_G : \{e, f\} \subseteq M'} w(M').$$

Proof. Let z be a fixed optimal dual solution. Since M is not a minimum-weight perfect matching, by Lemma 12, there are two cases: Either there exists an edge $e \in M$ with $e \notin E_z$ (in this case, take as f any edge from M and we are done), or there exists some odd set $A \in \mathcal{A}$ with $z_A > 0$, distinct edges $e, f \in M$, and both $e, f \in \delta(A)$. In the latter case, by the same lemma, no minimum-weight perfect matching contains both e and f . Hence, we get

$$\text{OPT} < \min_{M' \in \mathcal{M}_G : \{e, f\} \subseteq M'} w(M') = w(M)$$

where OPT denotes the weight of a minimum-weight perfect matching in G . ◀

It remains to use Lemma 13 repeatedly to obtain Theorem 9.

Proof of Theorem 9. We proceed by induction over ℓ . The case $\ell = 1$ is trivial and Lemma 13 already proves the case $\ell = 2$. Thus, let now $\ell \geq 3$ be arbitrary, and assume as induction hypothesis that the statement holds for all $\ell' < \ell$. Recall that M is an ℓ -th smallest perfect matching. We want to prove that there is a subset $F \subseteq E$ of edges of size at most $2(\ell - 1)$ such that

$$w(M) = \min_{M' \in \mathcal{M}_G : F \subseteq M'} w(M').$$

Let $M_{\min}$ be an arbitrary minimum-weight perfect matching of G . From the argument in Lemma 13, we know that there exist edges $e, f \in M$ such that

$$w(M_{\min}) < \min_{M' \in \mathcal{M}_G : \{e, f\} \subseteq M'} w(M') \leq w(M).$$

Consider now the graph G' obtained from G by removing e and f , their incident vertices, and all their incident edges. Observe that $M \setminus \{e, f\}$ is an ℓ' -th smallest perfect matching in G' for some $\ell' < \ell$. By the induction hypothesis, this means that there is a set $F' \subseteq E \setminus \{e, f\}$ of size at most $2(\ell' - 1)$ such that

$$w(M \setminus \{e, f\}) = \min_{M' \in \mathcal{M}_{G'} : F' \subseteq M'} w(M').$$

Define $F := F' \cup \{e, f\}$ and observe that

$$w(M) = w(\{e, f\}) + \min_{M' \in \mathcal{M}_{G'} : F' \subseteq M'} w(M') = \min_{M' \in \mathcal{M}_G : F \subseteq M'} w(M')$$

and that the set F has size at most $2(\ell - 1)$. ◀

4 Related Cycle Problems

In this section, we present the ideas behind Theorem 14 below. The full proof of Theorem 14 and the involved lemmata can be found in the Appendix of the full version.

► **Theorem 14** ($\text{EWPM} \equiv_p \text{ECS} \ \& \ \text{BCPM} \equiv_p \text{SOC}$). *EWPM is polynomial-time equivalent to ECS. BCPM is polynomial-time equivalent to SOC.*

As we will see, both equivalences use the same base construction and only differ in some minor configurations. The encoding of the weights is preserved in all reductions and hence ECS is equivalent to EM for weights encoded in unary, and NP-complete for weights encoded in binary.

From Matchings to Cycles

We start with the direction of $\text{EWPM} \leq_p \text{ECS}$ and $\text{BCPM} \leq_p \text{SOC}$, i.e. we give reductions starting from matching-type problems going to cycle-type problems. We use roughly the same construction for both reductions. We remark that in the case of matching-type problems in *bipartite graphs*, a similar reduction already exists, and it goes to cycle-type problems in *directed graphs* [27, 8]. In these existing reductions, the bipartition of the graph was used to define some edge orientation, which in turn ensured that cycles in the new graph would correspond to M -alternating cycles (where M is an arbitrary minimum-weight perfect matching) in the old graph (see [27] and [8] for more details). The same trick does not apply in general graphs, which is why we need to reduce to undirected cycle problems instead. In particular, the new trick will be to manipulate the weights such that the weight of any non-alternating cycle exceeds some threshold. The following lemma captures the key properties of this construction.

► **Lemma 15.** *Let $G = (V, E)$ be a graph with non-negative edge-weights $w : E \rightarrow \mathbb{N}$ that admits a minimum-weight perfect matching M . Consider the new edge-weights $w' : E \rightarrow \mathbb{Z}$ defined as*

$$w'(e) := \begin{cases} -w(e) - r - 1, & e \in M \\ w(e) + r + 1, & e \notin M \end{cases}$$

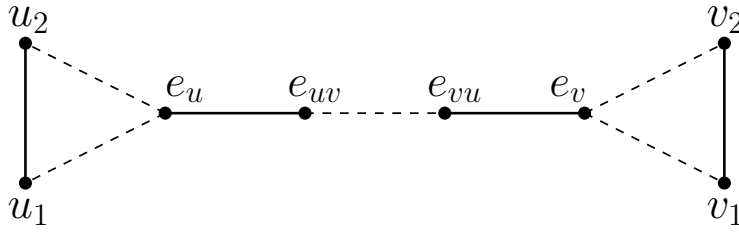
for some integer $r \geq w(M)$. Then every cycle C with $w'(C) \leq r - w(M)$ is M -alternating and w' is conservative.

Using Lemma 15, we can now informally describe the reductions: Assume first that we are given an EWPM-instance I consisting of $G = (V, E)$, $w : E \rightarrow \mathbb{Z}$, and $k \in \mathbb{Z}$. Without loss of generality, we can assume that the weights w are non-negative. We first compute a minimum-weight perfect matching M in G . Note that if no such matching exists or its weight exceeds k , I is a NO-instance and we can produce a simple NO-instance of ECS. Otherwise, we construct w' as in Lemma 15 (setting $r = k$) and return the ECS-instance I' consisting of G , w' and $k' := k - w(M)$. By Lemma 15, w' is conservative. Setting $r = k$ crucially guarantees that solutions to the ECS-instance can be translated back to solutions to the EWPM-instance I .

The only difference in the reduction from BCPM to SOC is that we need to manually check the parity of M and use a trivial construction if it already has correct parity.

From Cycles to Matchings

We will now turn our attention to the reverse reductions $\text{ECS} \leq_p \text{EWPM}$ and $\text{SOC} \leq_p \text{BCPM}$. The general idea is again inspired by the existing reductions between bipartite matching and directed cycle problems ([27, 8]): We want to duplicate every vertex and add edges between duplicates in order to ensure the existence of a canonical perfect matching M . Additionally, we want to ensure that the existence of any other perfect matching M' in the new graph implies existence of cycles in the original graph by looking at the symmetric difference $M \Delta M'$. Unfortunately, since our original graph is not directed anymore, we cannot use the ideas from [27, 8] to achieve this. Instead, we use the construction described in Lemma 16 and Figure 2.



■ **Figure 2** This figure shows the construction from Lemma 16 for a single edge $e = \{u, v\}$. Solid edges indicate the perfect matching M from Lemma 16. The edge $\{e_{uv}, e_{vu}\}$ receives the weight of e while all other edges get a weight of zero.

► **Lemma 16.** *Let $G = (V, E)$ be a graph with conservative edge-weights $w : E \rightarrow \mathbb{Z}$. We define*

$$\begin{aligned}
 V' &:= \{v_1, v_2 \mid v \in V\} \cup \{e_u, e_{uv}, e_{vu}, e_v \mid e = \{u, v\} \in E\} \\
 E' &:= \{\{v_1, v_2\} \mid v \in V\} \\
 &\quad \cup \{\{u_1, e_u\}, \{u_2, e_u\}, \{e_u, e_{uv}\}, \{e_{uv}, e_{vu}\}, \\
 &\quad \{e_{vu}, e_v\}, \{e_v, v_1\}, \{e_v, v_2\} \mid e = \{u, v\} \in E\} \\
 w' &:= e' \in E' \mapsto \begin{cases} w(e) & \text{if } e' = \{e_{uv}, e_{vu}\} \text{ for some } e = \{u, v\} \in E \\ 0 & \text{otherwise} \end{cases} \\
 M &:= \{\{v_1, v_2\} \mid v \in V\} \cup \{\{e_u, e_{uv}\}, \{e_{vu}, e_v\} \mid e = \{u, v\} \in E\}
 \end{aligned}$$

and consider the graph $G' = (V', E')$ with edge-weights w' and perfect matching M . Then there exist vertex-disjoint cycles $C_1, \dots, C_p$ of total weight k (w.r.t. w) in G if and only if there exist vertex-disjoint M -alternating cycles $C'_1, \dots, C'_p$ of total weight k (w.r.t. w') in G' .

Let us now briefly explain how this yields the desired reductions. Assume that we are given an ECS-instance I consisting of $G = (V, E)$, $w : E \rightarrow \mathbb{Z}$, and $k \in \mathbb{Z}$. We use the construction from Lemma 16 to obtain a graph G' with edge-weights w' and return the EWPM-instance I' consisting of $G' = (V', E')$, w' , and $k' := k$. The symmetric difference $M \Delta M'$ of M with any perfect matching M' in G' yields a set of vertex disjoint cycles that can be translated back to G using Lemma 16. The reduction from SOC to BCPM uses the same construction, but we need to additionally observe that we can get a single odd-weight cycle that translates through Lemma 16.

5 Final Remarks

In Section 3, we proved that we can deterministically find an ℓ -th smallest perfect matching in time $n^{\mathcal{O}(\ell)}$. This also allows us to solve certain instances of EWPM efficiently: Assume e.g. that we are given an EWPM instance with k such that $k_1 < k_2 < \dots < k_\ell = k$ are all the distinct weights up to k that perfect matchings in G can take. Then our algorithm can find a perfect matching of weight $k = k_\ell$ efficiently if ℓ is small.

Clearly, this also works in the case of BCPM. Moreover, our reductions from Section 4 also allow us to get similar guarantees for ECS and SOC, respectively. Indeed, assume that we are given an instance of ECS or SOC with k such that $k_1 < k_2 < \dots < k_\ell = k$ are all the distinct weights up to k that cycle sums in G can take. We apply our reduction to build a corresponding EWPM or BCPM instance. By Lemma 16, our reductions have a one-to-one correspondence between ℓ -th smallest perfect matchings and ℓ -th smallest cycle sums. Hence, the parameterized runtime guarantees of the SPM_ℓ -algorithm carry through the reduction.

► **Corollary 17.** *BCPM can be solved in time $n^{\mathcal{O}(\ell)}$ if $k_1 < k_2 < \dots < k_\ell = k$ are all the distinct weights up to k that perfect matchings in G can take. ECS and SOC can be solved in time $n^{\mathcal{O}(\ell)}$ if $k_1 < k_2 < \dots < k_\ell = k$ are all the distinct weights up to k that cycle sums in G can take.*

References

- 1 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, Cambridge ; New York, 2009.
- 2 Chandra R. Chegireddy and Horst W. Hamacher. Algorithms for finding k-best perfect matchings. *Discrete applied mathematics*, 18(2):155–165, 1987. doi:10.1016/0166-218X(87)90017-5.
- 3 Anita Dür, Nicolas El Maalouly, and Lasse Wulf. An Approximation Algorithm for the Exact Matching Problem in Bipartite Graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, volume 275 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:21, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.18.
- 4 Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69:125–130, 1965.
- 5 Nicolas El Maalouly. Exact Matching: Algorithms and Related Problems. In *40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.STACS.2023.29.
- 6 Nicolas El Maalouly, Sebastian Haslebach, and Lasse Wulf. On the Exact Matching Problem in Dense Graphs. In *41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.STACS.2024.33.
- 7 Nicolas El Maalouly and Raphael Steiner. Exact Matching in Graphs of Bounded Independence Number. In *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.MFCS.2022.46.
- 8 Nicolas El Maalouly, Raphael Steiner, and Lasse Wulf. Exact Matching: Correct Parity and FPT Parameterized by Independence Number. In *34th International Symposium on Algorithms and Computation (ISAAC 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ISAAC.2023.28.
- 9 Nicolas El Maalouly and Lasse Wulf. Exact Matching and the Top-k Perfect Matching Problem, September 2022. doi:10.48550/arXiv.2209.09661.
- 10 David Eppstein. Representing all minimum spanning trees with applications to counting and generation. Technical report, UC Irvine: Donald Bren School of Information and Computer Sciences, 1995. URL: <https://escholarship.org/uc/item/5w35f33b>.
- 11 David Eppstein. K-best enumeration. *Bulletin of EATCS*, 1(115), 2015.
- 12 Anna Galluccio and Martin Loebl. On the Theory of Pfaffian Orientations. I. Perfect Matchings and Permanents. *The Electronic Journal of Combinatorics*, pages R6–R6, 1999. doi:10.37236/1438.
- 13 Jim Geelen and Rohan Kapadia. Computing Girth and Cogirth in Perturbed Graphic Matroids. *Combinatorica*, 38(1):167–191, February 2018. doi:10.1007/s00493-016-3445-3.
- 14 Rohit Gurjar, Arpita Korwar, Jochen Messner, Simon Straub, and Thomas Thierauf. Planarizing Gadgets for Perfect Matching Do Not Exist. In *Mathematical Foundations of Computer Science 2012*, pages 478–490, Berlin, Heidelberg, 2012. Springer. doi:10.1007/978-3-642-32589-2_43.
- 15 Meike Hatzel, Konrad Majewski, Michał Pilipczuk, and Marek Sokołowski. Simpler and faster algorithms for detours in planar digraphs. In *2023 Symposium on Simplicity in Algorithms (SOSA)*, Proceedings, pages 156–165. Society for Industrial and Applied Mathematics, January 2023. doi:10.1137/1.9781611977585.ch15.

- 16 Xinrui Jia, Ola Svensson, and Weiqiang Yuan. The Exact Bipartite Matching Polytope Has Exponential Extension Complexity. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 1635–1654. Society for Industrial and Applied Mathematics, January 2023. doi:10.1137/1.9781611977554.ch61.
- 17 Donald B. Johnson. Efficient Algorithms for Shortest Paths in Sparse Networks. *J. ACM*, 24(1):1–13, January 1977. doi:10.1145/321992.321993.
- 18 Kuo-Hua Kao, Jou-Ming Chang, Yue-Li Wang, and Justie Su-Tzu Juan. A Quadratic Algorithm for Finding Next-to-Shortest Paths in Graphs. *Algorithmica*, 61(2):402–418, October 2011. doi:10.1007/s00453-010-9402-4.
- 19 A. V. Karzanov. Maximum matching of given weight in complete and complete bipartite graphs. *Cybern Syst Anal*, 23(1):8–13, January 1987. doi:10.1007/BF01068796.
- 20 Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer, Berlin, Heidelberg, 2018. doi:10.1007/978-3-662-56039-6.
- 21 I. Krasikov and S. D. Noble. Finding next-to-shortest paths in a graph. *Information Processing Letters*, 92(3):117–119, November 2004. doi:10.1016/j.ipl.2004.06.020.
- 22 Kumar N. Lalgudi and Marios C. Papaefthymiou. Computing strictly-second shortest paths. *Information Processing Letters*, 63(4):177–181, August 1997. doi:10.1016/S0020-0190(97)00122-1.
- 23 Shisheng Li, Guangzhong Sun, and Guoliang Chen. Improved algorithm for finding next-to-shortest paths. *Information Processing Letters*, 99(5):192–194, September 2006. doi:10.1016/j.ipl.2006.04.013.
- 24 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, March 1987. doi:10.1007/BF02579206.
- 25 Hitoshi Murakami and Yutaro Yamaguchi. An FPT Algorithm for the Exact Matching Problem and NP-Hardness of Related Problems. *IEICE Transactions on Information and Systems*, E108.D(3):214–220, March 2025. doi:10.1587/transinf.2024FCP0009.
- 26 Katta G Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations research*, 16(3):682–687, 1968.
- 27 Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of restricted spanning tree problems. *J. ACM*, 29(2):285–309, April 1982. doi:10.1145/322307.322309.
- 28 Ildikó Schlotter and András Sebő. Odd Paths, Cycles, and T-Joins: Connections and Algorithms. *SIAM Journal on Discrete Mathematics*, 39(1):484–504, March 2025. doi:10.1137/23M158156X.
- 29 J. W. Suurballe. Disjoint paths in a network. *Networks*, 4(2):125–145, January 1974. doi:10.1002/net.3230040204.
- 30 Vijay V. Vazirani. NC algorithms for computing the number of perfect matchings in K₃,3-free graphs and related problems. In *SWAT 88*, pages 233–242, Berlin, Heidelberg, 1988. Springer. doi:10.1007/3-540-19487-8_27.
- 31 Bang Ye Wu. A Simpler and More Efficient Algorithm for the Next-to-Shortest Path Problem. *Algorithmica*, 65(2):467–479, February 2013. doi:10.1007/s00453-011-9601-7.
- 32 Raphael Yuster. Almost Exact Matchings. *Algorithmica*, 63(1-2):39–50, June 2012. doi:10.1007/S00453-011-9519-0.
- 33 Cong Zhang and Hiroshi Nagamochi. The Next-to-Shortest Path in Undirected Graphs with Nonnegative Weights. *Theory of Computing*, 128, 2012. URL: <http://crpit.scem.westernsydney.edu.au/abstracts/CRPITV128Zhang.html>.