On the Exact Matching Problem in Dense Graphs

Nicolas El Maalouly
Department of Computer Science, ETH Zurich, Switzerland

Sebastian Haslebacher
Department of Computer Science, ETH Zurich, Switzerland

Lasse Wulf
Institute of Discrete Mathematics, TU Graz, Austria

Abstract

In the Exact Matching problem, we are given a graph whose edges are colored red or blue and the task is to decide for a given integer k, if there is a perfect matching with exactly k red edges. Since 1987 it is known that the Exact Matching Problem can be solved in randomized polynomial time. Despite numerous efforts, it is still not known today whether a deterministic polynomial-time algorithm exists as well. In this paper, we make substantial progress by solving the problem for a multitude of different classes of dense graphs. We solve the Exact Matching problem in deterministic polynomial time for complete r-partite graphs, for unit interval graphs, for bipartite unit interval graphs, for graphs of bounded neighborhood diversity, for chain graphs, and for graphs without a complete bipartite t-hole. We solve the problem in quasi-polynomial time for Erdős-Rényi random graphs $G(n, 1/2)$. We also reprove an earlier result for bounded independence number/bipartite independence number. We use two main tools to obtain these results: A local search algorithm as well as a generalization of an earlier result by Karzanov.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Exact Matching, Perfect Matching, Red-Blue Matching, Bounded Color Matching, Local Search, Derandomization

Digital Object Identifier 10.4230/LIPIcs.STACS.2024.33

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2401.03924`

Funding *Lasse Wulf*: Supported by the Austrian Science Fund (FWF): W1230.

1 Introduction

A fundamental problem in computer science is the question whether a randomized algorithm has any sort of advantage over a deterministic algorithm. In particular, theoretical computer scientists are concerned with the question: $\mathsf{P} = \mathsf{BPP}$? Here, P contains decision problems that can be solved deterministically in polynomial-time, while BPP contains decision problems that can be solved with randomized algorithms in polynomial-time (under a two-sided, bounded error probability [2]). One can also define the classes $\mathsf{RP} \subseteq \mathsf{BPP}$ and $\mathsf{CoRP} \subseteq \mathsf{BPP}$ of randomized polynomial-time algorithms with one-sided error probability (the difference between the two classes is the side of the error). Nowadays, experts in complexity theory believe that $\mathsf{P} = \mathsf{RP} = \mathsf{CoRP} = \mathsf{BPP}$, i.e. it is believed that randomness does not offer any sort of advantage for the task of solving a problem in polynomial time. The reason for this belief are deep connections between complexity theory, circuit lower bounds, and pseudorandom generators [2, 24, 26].

While it would be intriguing to attack the conjecture $\mathsf{P} = \mathsf{BPP}$ directly, it seems very hard to make direct progress in this way. In particular, $\mathsf{P} = \mathsf{BPP}$ would imply deterministic algorithms for *all* problems which can be solved with randomness. A more humble approach

41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024).
Editors: Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov; Article No. 33; pp. 33:1–33:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

one can take is to look for one specific problem, where the research community knows a randomized, but no deterministic algorithm, and try to find a deterministic algorithm for this specific problem. Every single of these results can be seen as further evidence towards $\mathsf{P} = \mathsf{BPP}$. One famous example of such a "derandomization" is the deterministic algorithm for primality testing by Agrawal, Kayal and Saxena [1] from 2002.

Quite interestingly, we only know of a handful of problems where a randomized but no deterministic polynomial-time algorithm is known. This paper is concerned with one of these examples, the *Exact Matching problem* (EM). Given an integer k and a simple graph G together with a coloring of its edges in red or blue, EM is the problem of deciding whether G has a perfect matching with exactly k red edges. EM was introduced by Papadimitriou and Yannakakis [36] back in 1982. Not too long after its introduction, in 1987, Mulmuley et al. [33] showed that EM can be solved in randomized polynomial-time. Despite the original problem being from 1982, and in spite of multiple applications of EM in different areas (see next paragraph), it is still not known today, if a deterministic polynomial-time algorithm exists.

Another interesting aspect of EM is its connection to *polynomial identity testing* (PIT). PIT is another one of the rare problems in BPP for which we still do not know any deterministic polynomial-time algorithm. Given a multivariate polynomial described by an algebraic circuit, PIT is the problem of deciding whether the polynomial is identically equal to zero or not. Using the well-known Schwartz-Zippel Lemma (named after Schwartz [37] and Zippel [44] who discovered it in the eighties), it is clear that PIT belongs to CoRP. Therefore, under the conjecture $\mathsf{CoRP} = \mathsf{P}$, there should be a deterministic polynomial-time algorithm for PIT. However, Kabanets and Impagliazzo [26] provided strong evidence that derandomizing PIT might be notoriously hard, since it would imply proving circuit lower bounds. The known randomized algorithm for EM uses PIT as a subroutine on a slightly modified Tutte matrix of the given graph. Alternatively, one can substitute the use of PIT with the famous Isolation Lemma due to Mulmuley et al. [33]. Both approaches lead to randomized polynomial-time algorithms for EM and show that EM is contained in the class RP.

History of Exact Matching

We have already established that EM should belong to P if we believe the conjecture $\mathsf{P} = \mathsf{RP} = \mathsf{BPP}$. However, the best deterministic algorithm to date takes exponential time. This is especially astonishing knowing that EM was introduced by Papadimitriou and Yannakakis [36] back in 1982. A few years later, in 1987, Mulmuley et al. [33] showed that EM can be solved in randomized polynomial-time in their famous paper that also introduced the Isolation Lemma. In fact, their algorithm additionally allows for a high degree of parallelism i.e. they proved that EM belongs to RNC (and hence also to RP and BPP). RNC is defined as the class of decision problems allowing an algorithm running in polylogarithmic time using polynomially many parallel processors, while having additional access to randomness (we refer the interested reader to [6, Chapter 12] for a formal definition). This means that if we allow for randomness, EM can be solved efficiently even in parallel, while the best known deterministic algorithm requires exponential time.

In the same year 1987, Karzanov [27] gave a precise characterization of the solution landscape of EM in complete and complete bipartite graphs. His characterization also implies deterministic polynomial-time algorithms for EM restricted to those graph classes. Several articles later appeared [19, 23, 42], simplifying and restructuring those results.

EM is known to admit efficient deterministic algorithms on some other restricted graph classes as well: With standard dynamic programming techniques, EM can be solved in polynomial-time on graphs of bounded tree-width [11, 40]. Moreover, derandomization results exist for $K_{3,3}$-minor free graphs [41, 43] and graphs of bounded genus [18]. These works make use of so-called Pfaffian orientations. Besides solving EM on restricted graph classes, some prior work has also focused on solving EM approximately. Yuster [43] proved that in a YES-instance, we can always find an almost exact matching in deterministic polynomial-time (an almost exact matching is a matching with exactly k red edges that fails to cover only two vertices).

This completes a summary of the history of the EM problem until two years ago. With so little progress, one might wonder if the community has lost interest in, or forgot about the problem. However, over the last decade alone, we have seen the problem appear in the literature from several areas. This includes budgeted, color bounded, or constrained matching [5,28,31,32,38], multicriteria optimization [21], matroid intersection for represented matroids [7], binary linear equation systems [4], recoverable robust assignment [17], or planarizing gadgets for perfect matchings [22]. In many of these papers, a full derandomization of EM would also derandomize some or all of the results of the paper. EM also appeared as an interesting open problem in the seminal work on the parallel computation complexity of the matching problem [39], which might be partly responsible for the increase in attention that the problem has received recently.

Recent progress

As mentioned above, until recently, EM was only solved for a handful of graph classes. This is even more extreme in the case of dense graphs where it was only solved on complete and complete bipartite graphs. In 2022, El Maalouly and Steiner [12] finally made progress on this side by showing that EM can be solved on graphs of bounded *independence number* and bipartite graphs of bounded *bipartite independence number*. Here, the independence number of a graph G is defined as the largest number α such that G contains an *independent set* of size α. The bipartite independence number of a bipartite graph G equipped with a bipartition of its vertices is defined as the largest number β such that G contains a *balanced independent set* of size 2β, i.e., an independent set using exactly β vertices from both color classes. This generalizes previous results for complete and complete bipartite graphs, which correspond to the special cases $\alpha = 1$ and $\beta = 0$. The authors also conjectured that counting perfect matchings is #P-hard for this class of graphs. This conjecture was later proven in [14] already for $\alpha = 2$ or $\beta = 3$. This makes them the first classes of graphs where EM can be solved, even though counting perfect matchings is #P-hard. This work was later extended to an FPT-algorithm on bipartite graphs parameterized by the bipartite independence number [12].

There has also been a recent interest in approximation algorithms for EM. Such approximation algorithms have been developed for the closely related *budgeted matching* problem, where sophisticated methods were used to achieve a PTAS [5] and, more recently, an efficient PTAS [8]. These methods however do not guarantee to return a perfect matching (but note that a deterministic FPTAS for budgeted matching would imply a deterministic polynomial-time algorithm for EM [5]). In [9,11] it is argued that relaxing the perfect matching constraint takes away most of the difficulty of the problem. In contrast, the aim of the recent work has been to keep the perfect matching constraint and relax the requirement on the number of red edges. The first such result was given in [12], where it was shown that in a bipartite graph we can always find a perfect matching with at least $0.5k$ and at most $1.5k$ red edges in deterministic polynomial-time. This represents a two-sided approximation for the problem. Shortly after, [9] studied the surprisingly much more difficult problem of getting a one-sided approximation and presented a 3-approximation in that setting (i.e. an algorithm that outputs a perfect matching with at least $k/3$ and at most k red edges), relying on a newly defined concept of graph rigidity.

Another relaxation of the problem is to consider only modular constraints on the number of red edges, e.g., requiring the output perfect matching to have an odd number of red edges. In the case of bipartite graphs, the problem can be solved using the more general result of [3] on network matrices. This does not work for general graphs, for which the problem was solved in [13] with a different approach relying on a deep result by Lovász [30] on the *linear hull* of perfect matchings. The problem remains open for other congruency constraints, e.g., requiring the output to have (r mod p) red edges for some integers r and p. The latter problem has been used by [34] as a building block in an algorithm for a special class of integer programs having a constraint matrix with bounded subdeterminants. This means that a deterministic algorithm for this special case of EM would also derandomize the algorithm of [34].

In [11], the Top-k Perfect Matching problem is introduced, where the input is a weighted graph and the goal is to find a PM that maximizes the weight of the k heaviest edges in the matching. In combination with the result from [15], the problem is shown to be polynomially equivalent to EM when the input weights are polynomially bounded. Several approximation and FPT algorithms were also developed.

Another recent line of work follows a polyhedral approach to understand the differences between finding a perfect matching and EM [25]. In particular, the authors show exponential extension complexity for the bipartite exact matching polytope. This stands in contrast with the bipartite perfect matching polytope whose vertices are all integral [10].

Finally, [40] studies some generalizations of EM to matching problems with vertex color constraints and shows an interesting connection to quantum computing.

Our contribution

In this paper, we study EM on dense graph classes. We are able to solve EM in deterministic polynomial time on many different classes of dense graphs, which before could only be handled by a randomized algorithm. In order to achieve this result, we use two key techniques: First, a local search algorithm, second, a generalization of Karzanov's [27] theorem. With the first technique, the local search algorithm, we obtain the following results: (For a formal definition of all the graph classes listed, as well as a motivation for why we consider exactly these classes, we refer the reader to Section 2.1.)

- There is a deterministic $n^{O(1)}$ time algorithm for EM on complete r-partite graphs for all $r \geq 1$. The constant in the exponent is independent of r. This is an extension of the special cases $r = n$ and $r = 2$, which correspond to the cases of complete and complete bipartite graphs [27] already known in 1987.
- There is a deterministic $n^{O(1)}$ time algorithm for EM on graphs of bounded neighborhood diversity $d = O(1)$. The neighborhood diversity is a parameter popular in the area of parameterized complexity [29].
- There is a deterministic $n^{O(1)}$ time algorithm for EM on graphs G which have no complete bipartite t-hole (i.e. $K_{t,t} \not\subseteq \overline{G}$) with $t = O(1)$.
- There is a deterministic $n^{O(\log^{12}(n)p^{-12})}$ time algorithm for EM on the random graph $G(n, p)$. By this, we mean the following: We say an algorithm is *correct* for a graph G, if for all possible red-blue edge colorings of G and all possible k, the algorithm correctly solves EM on that input. We show that there is a deterministic algorithm $\mathcal{A}$, which always halts in $n^{O(\log^{12}(n)p^{-12})}$ steps, and if G is sampled from the distribution $G(n, p)$, then with high probability $\mathcal{A}$ is correct for G. As a special case, we obtain a quasi-polynomial algorithm for $G(n, 1/2)$. We are the first authors to consider EM from the perspective of random graphs.

- As a special case, our main theorem contains a re-proof of the two main results of [12], showing that there is a deterministic $n^{O(1)}$ algorithm for EM on graphs of bounded independence number/ bip. graphs of bounded bip. independence number. Our result is therefore a large generalization of this earlier result and puts it into the bigger context of local search.

We also identify a certain graph property, which we call the *path-shortening property*. Graphs which are very dense and structured are candidates to examine for this property. Our main theorem is that for every graph with the path-shortening property, a local search approach can be used to correctly solve the Exact Matching Problem. In fact, all the examples above follow from our main theorem. We remark that our local search algorithm is very simple, only the proof of its correctness is quite involved. The main idea of the observation is that in graphs with the path-shortening property, strong locality statements about the set of all perfect matchings can be made. Details are presented in Section 3.

While the local search approach allows us to tackle several new graph classes, we still notice that it fails even on some very dense and structured graph classes. In particular, we are interested in graph classes which are related to the problem of counting perfect matchings (for example, Okamoto et al. [35] list chordal, interval, unit interval, bipartite chordal, bipartite interval, and chain graphs among others). We highlight one example, the case of so-called chain graphs, where our local search fails.

This failure inspires us to seek other methods to understand the EM problem on dense graphs and leads us to consider our second key technique. We call this technique *Karzanov's property*, as it is a generalization of the result by Karzanov [27]. We show that several graph classes have Karzanov's property, including classes where our local search algorithm fails. We introduce a related property, which we call the *chord property*. We introduce a novel binary-search like procedure, which gives us both an efficient algorithm for EM on these graph classes, as well as a characterization of their solution landscape.

Finally, we are also able to identify some graph classes, where Karzanov's property almost holds. We call this the *weak Karzanov's property*. For those graph classes, we are not able to solve EM deterministically, but we are at least able to show that one can always find a PM with either k or $k-1$ red edges. Hence we can come very close to solving the problem. These graph classes are therefore obvious candidates to attack next in the effort of derandomizing EM. In summary, we obtain the following.

- There is a deterministic $n^{O(1)}$ time algorithm for EM on chain graphs, unit interval graphs, bipartite unit interval graphs and complete r-partite graphs for all $r \geq 1$. The solution landscape for these graph classes can also be characterized by the perfect matchings of maximum and minimum number of red edges with a given parity.
- There is a deterministic $n^{O(1)}$ time algorithm on interval graphs, bipartite interval graphs, strongly chordal graphs and bipartite chordal graphs, that outputs a perfect matching with either $k-1$ or k red edges or deduces that the answer of the given EM-instance is "No".

Organization of the paper

In the following we start with some preliminaries in Section 2. Then in Section 3 we introduce our local search algorithm, the main ideas behind its correctness and its limitations. In Section 4 we introduce Karzanov's property, as well as Karzanov's weak property and discuss the main ideas behind their utility and limitations.

In the appendix of the full version of this paper, you can find the detailed proofs missing from Section 3, proofs that the local search algorithm works for several graph classes, the detailed proofs missing from Section 4, and proofs that several graph classes satisfy Karzanov's property while some others only satisfy Karzanov's weak property.

2 Preliminaries and Problem Definition

All graphs in this paper are undirected and simple. For a graph $G = (V, E)$, we denote by $V(G) := V$ its vertex set and by $E(G) := E$ its edge set. We usually use the letters n, m to denote $n := |V|$ and $m := |E|$. In this paper, paths and cycles are always simple (i.e. no vertex is repeated). In order to simplify the notation, we identify paths and cycles with their edge sets. Any reference to their vertices will be made explicit. The *neighborhood* $N(v)$ of a vertex v is the set of all vertices adjacent to v. A *colored graph* in this paper is a graph where every edge has exactly one of two colors, i.e. a tuple (G, c) with $c : E(G) \to \{\text{red}, \text{blue}\}$. For a subset $F \subseteq E$ of edges, we denote by $R(F) := \{e \in F \mid e \text{ is red}\}$ its set of red edges and by $r(F) := |R(F)|$ its number of red edges. Analogously, we define $B(F)$ and $b(F)$ for blue edges. A *matching* of G is a set $M \subseteq E$ of edges, which touches every vertex at most once. A *perfect matching* (abbreviated PM) is a matching M which touches every vertex exactly once. An edge e is called *matching*, if $e \in M$ and *non-matching* otherwise. The *Exact Matching Problem* is formally defined as follows.

Problem EM
Input: Colored graph (G, c), integer $k \geq 0$.
Question: Is there a perfect matching M in G such that $r(M) = k$?

The *symmetric difference* $A \triangle B$ of two sets A and B is $A \cup B \setminus (A \cap B)$. Let M be a PM. An *M-alternating cycle*, or simply an *alternating cycle* (if M is clear from context), is a cycle which alternates between edges in M and edges not in M. An *alternating path* is defined analogously. If M_1, M_2 are PMs, it is well known that $D := M_1 \triangle M_2$ is a vertex-disjoint union of alternating cycles. Since $\triangle$ behaves like addition mod 2, we also have $M_2 = M_1 \triangle D$ and $M_1 = M_2 \triangle D$. In this paper, we try to follow the convention that the letter C denotes a single cycle and the letter D denotes a vertex-disjoint union of one or more cycles.

2.1 Definition of Graph Classes

Throughout the paper, we show how to solve EM on various classes of dense graphs. In this subsection, we properly define all graph classes used.

The motivation to consider exactly those classes comes from different sources. Some of the classes considered are direct generalizations of classes, where it was previously known that EM can be solved. Other classes are generally well-known. For the remaining classes, we regard them as interesting, because they appear in the context of counting the number of perfect matchings. (For example, in their paper about counting perfect matchings, Okamoto et al. [35] list chordal, interval, unit interval, bipartite chordal, bipartite interval, and chain graphs among others.) The reason for this is, that if it is #P-hard to count the number of perfect matchings, then the Pfaffian derandomization method used in [18, 41, 43] is unlikely to work (compare [15] for details). We also pay special attention to bipartite graphs, since we expect EM to be easier to tackle if the graph is bipartite.

A graph is *complete r-partite*, if the vertex set can be partititioned into r parts $V_1, \dots, V_r$ such that inside each part there are no edges, and between two different parts, there are all the possible edges. The case $r = n$ corresponds to the complete graph, while $r = 2$ corresponds to the complete bipartite graph. A generalization of complete r-partite graphs, if $r = O(1)$, are graphs of *bounded neighborhood diversity*, a parameter coming from parameterized complexity [29]. A graph has neighborhood diversity d, if $V(G)$ can be partitioned into d parts $V_1 \dots, V_d$, such that between every two parts V_i, V_j with $i \neq j$, there are either no edges, or all the possible edges, and every part itself induces either a complete or an empty graph.

The Erdős-Rényi random graph $G(n,p)$ is the random graph on n vertices, where every edge appears with probability p independently [16]. It is very well-studied and has a rich history.

The remaining definitions in this subsection are motivated by [35]. Furthermore, many of these graph classes are extensively studied in algorithmic graph theory [20]. A graph $G = (V, E)$ is an *interval graph* if there exists a mapping $I : V \to \{[a,b] \subseteq \mathbb{R} \mid a \leq b\}$ such that $\{u, v\} \in E \iff I(u) \cap I(v) \neq \emptyset$ holds for all distinct $u, v \in V$. If additionally $I(v)$ is a unit interval for all vertices v, then G is called a *unit interval graph.*

We may consider bipartite versions of interval graphs the following way: A bipartite graph $G = (X \dot\cup Y, E)$ is a *bipartite interval graph* if there exists a mapping $I : X \dot\cup Y \to \{[a,b] \subseteq \mathbb{R} \mid a \leq b\}$ such that $\{x, y\} \in E \iff I(x) \cap I(y) \neq \emptyset$ holds for all $x \in X, y \in Y$. If additionally $I(v)$ is a unit interval for all vertices v, then G is called a *bipartite unit interval graph.* Note that by this definition, if two vertices $x, y \in X$ are in the same color class of the bipartition, there is no edge between x and y even if the intervals $I(x)$ and $I(y)$ intersect.

Interval graphs are *strongly chordal.* A graph G is called strongly chordal if every cycle of length at least 4 admits a chord and every even cycle of length at least 6 admits an odd chord (i.e. a chord that splits the cycle into two odd length paths).

Bipartite interval graphs are *bipartite chordal.* A bipartite graph G is bipartite chordal if and only if every cycle of (necessarily even) length at least 6 admits a chord.

Finally, we consider a special case of bipartite interval graphs, so-called *chain graphs.* A bipartite graph $G = (X \dot\cup Y, E)$ is a chain graph if and only if its vertices can be relabeled as $x_1, \dots, x_{|X|} \in X$ and $y_1, \dots, y_{|Y|} \in Y$ such that $N(x_i) \subseteq N(x_{i+1})$ and $N(y_j) \subseteq N(y_{j+1})$ hold for all $1 \leq i < |X|$ and $1 \leq j < |Y|$.

3 Local Search

As of course is well known, the central idea behind a local search algorithm is to only examine solutions close to the current solution at every step. Hence we require a notion of distance. For our purpose, this notion is as follows.

▶ **Definition 1.** *Let (G, c) be a colored graph and $M_1, M_2 \subseteq E(G)$ be two PMs. The distance between M_1, M_2 is*

$$\operatorname{dist}(M_1, M_2) := \min\{r(M_1 \triangle M_2),\ b(M_1 \triangle M_2).\}$$

For an integer $s \geq 0$, the s-neighborhood of a PM M is

$$\mathcal{N}_s(M) := \{M' \subseteq E(G) \mid M' \textit{ is a PM},\ \operatorname{dist}(M, M') \leq s\}.$$

Note that $\operatorname{dist}(M_1, M_2) = \min\{|R(M_1) \triangle R(M_2)|,\ |B(M_1) \triangle B(M_2)|\}$. In other words, two PMs have small distance if and only if their two sets of red edges are almost the same, or their two sets of blue edges are almost the same. For example, if two PMs have the same set of red edges, i.e. $R(M_1) = R(M_2)$, then $\operatorname{dist}(M_1, M_2) = 0$, even if their set of blue edges is completely different.

Observe that as a consequence of this definition, for a fixed PM M even the 0-neighborhood $\mathcal{N}_0(M)$ may have exponential size in n. This is a problem for us: how can we perform local search, if the size of the neighborhood is exponential? Fortunately, there is a fix: We do not need to know the complete neighborhood of M, all we need to know is which values of $r(M')$ are possible to achieve in the neighborhood, i.e. the set of all k' such that there exists a PM M' in the neighborhood with $r(M') = k'$. The following lemma states that this information can be computed efficiently. The idea is to guess either the set $R(M')$ or the set $B(M')$ and see if the guess can be completed to a PM using only edges of the opposite color.

▶ **Lemma 2.** *Assume we are given a PM M in a colored graph, and an integer $s \geq 0$. There is an algorithm which runs in $O(m^{s+3})$ time and computes the set $\{k' \in \mathbb{N} \mid \exists M' \in \mathcal{N}_s(M), r(M') = k'\}$ and for each k' in this set outputs at least one representative M'' with $r(M'') = k'$.*

Proof. Let (G, c) be the colored graph with $G = (V, E)$, and let $E_R := R(E)$ be the set of all red edges and $E_B := B(E)$ be the set of all blue edges. The algorithm works as follows:

1. Enumerate all (not necessarily perfect) matchings $X \subseteq E_R$ with $|X \triangle R(M)| \leq s$. For each such X, use a classical maximum matching algorithm on the blue edges to check whether there exists $Y \subseteq E_B$ such that $X \dot\cup Y =: M''$ is a PM. If the answer is affirmative, we add the number $k' := |X| = r(M'')$ to the output set (together with its representative M'').
2. After that, we repeat the same procedure with the colors switched: Enumerate all matchings $X \subseteq E_B$ with $|X \triangle B(M)| \leq s$. For each such X, check whether there exists $Y \subseteq E_R$ such that $X \dot\cup Y$ is a PM. If yes, we add the number $k' := n/2 - |X|$ to the output set (together with its representative M'').

The enumeration of sets X can be done in $O(m^s)$ time. Note that this algorithm is sound, in the sense that every PM M'' generated by it is indeed contained in $\mathcal{N}_s(M)$. On the other hand, the algorithm is also complete: If $M' \in \mathcal{N}_s(M)$, then either $R(M')$ or $B(M')$ appears in the enumeration. This means that not necessarily M', but at least some M'' with $r(M') = r(M'')$ is found by the algorithm. The total runtime of the algorithm is $\Theta(m^s f_M)$, where f_M denotes the time it takes to solve the perfect matching problem deterministically. For simplification, we let $f_M = O(mn^2) = O(m^3)$ [10]. ◀

Algorithm 1 A simple local search algorithm, Local(s).

```
Input: Colored graph (G, c), integer k ≥ 0, local search parameter s ≥ 0
Result: Either a PM M with r(M) = k, or the info that local search was
        unsuccessful.
M_min ←PM in G with minimum number of red edges among all PMs ;
M ← M_min;
while r(M) ≠ k do
    Try to find M' ∈ N_s(M) s.t. r(M) < r(M') ≤ k using Lemma 2;
    if successful then
        M ← M';
    else
        return "local search failed.";
return M;
```

With Lemma 2 in mind, we introduce Algorithm 1 as the most natural local search algorithm. It starts with a PM with the minimum number of red edges and iteratively tries to increase $r(M)$. Note that the PM $M_{\min}$ in the first line of the algorithm can be computed in polynomial time (one can run a classical maximum weight perfect matching algorithm, where red edges receive weight -1, and blue edges receive weight 0). Algorithm 1 can return false negatives, in the sense that given a yes-instance of EM it is possible for the algorithm to get stuck in a local optimum and return "false". Algorithm 1 can not return false positives. If we increase the search parameter s, we expect Algorithm 1 to be correct more often on average, but we also expect a longer runtime. We denote Algorithm 1 with parameter s by the name LOCAL(s). Since every successful iteration increases $r(M)$, the running time of LOCAL(s) is bounded by $O(m^{s+4})$. It is desirable to understand when LOCAL(s) correctly solves EM. This is partially answered in the next subsection.

3.1 A Sufficient Condition for Local Search

We present a sufficient condition for LOCAL(s) to correctly solve EM. Although the algorithm LOCAL(s) is quite simple, the proof that our condition suffices for correctness of the algorithm is involved. The main idea is the observation that in certain dense and highly structured graphs, it is possible to prove strong locality properties for the set of all perfect matchings. In particular, we consider graphs which have the following technical property:

▶ **Definition 3.** *Let $t \geq 2$ be an integer. A graph G has the so-called path-shortening property* PSHORT(t)*, if for all PMs $M \subseteq G$, and for all M-alternating paths P the following holds: If $F \subseteq P \cap M$ is a subset of matching edges of size $|F| = t$ and $F = \{\{a_1, b_1\}, \dots, \{a_t, b_t\}\}$, where the vertices $a_1, b_1, a_2, b_2, \dots, a_t, b_t$ appear in this order along the path, then the graph G contains an edge $\{a_i, b_j\}$ for some indices $1 \leq i < j \leq t$ or both the edges $\{a_{i_1}, a_{i_3}\}, \{b_{i_2}, b_{i_4}\}$ for some indices $1 \leq i_1 < i_2 < i_3 < i_4 \leq n$.*

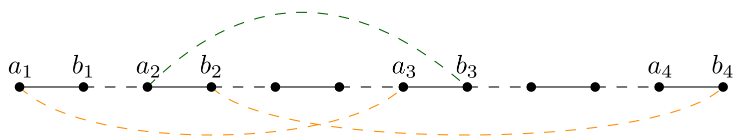


Figure 1 An example of the property PSHORT(4) on a path of length 11. Matching edges are bold. Both possibilities of path shortening are highlighted.

An illustration of this property is provided in Figure 1. Note that the property is monotone in t, i.e. PSHORT(t) implies PSHORT(t') for all $t' > t$. Our main result is the insight that the property PSHORT is sufficient for local search to be correct.

▶ **Theorem 4.** *If a graph G has property PSHORT(t), then the deterministic algorithm LOCAL($O(t^{12})$) solves EM on graph G in time $n^{O(t^{12})}$ (for all possible edge colorings $c : E(G) \to \{red, blue\}$ and all target values $k \in \mathbb{N}$).*

In particular, if $t = O(1)$, then the algorithm above is polynomial-time. Such a theorem is only useful of course, if we can show that many different graphs have this mysterious property PSHORT. Indeed, we can show (proofs can be found in the appendix of the full version of this paper):

- Complete r-partite graphs have the property PSHORT(3) for all integers $r \geq 1$.
- Graphs of bounded neighborhood diversity d have the property PSHORT($d+1$).
- Graphs of bounded independence number α have the property PSHORT($2\,\mathrm{Ram}(\alpha+1)$), where $\mathrm{Ram}(x) \leq 4^x$ is the diagonal Ramsey number.
- Graphs of bounded bipartite independence number β have property PSHORT($2\beta+2$).
- If a graph G has no complete bipartite t-hole (i.e. the complement $\overline{G}$ does not contain $K_{t,t}$ as subgraph), then G has the property PSHORT($2t$).
- The random graph $G(n,p)$ has property PSHORT($2\log(n)/p$) with high probability.

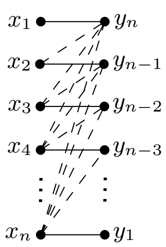
The proof of Theorem 4 is quite technical and requires many steps. The complete proof is given in the full version of this paper. The main insight is the observation that in graphs with property PSHORT, the set of all perfect matchings must obey strong locality guarantees. For the proof of this locality statement, we introduce the new idea of *local modifiers*. Each local modifier has a weight associated to it, and the goal becomes to combine the weights in such a way, that they cancel out to be 0. The proof uses ideas and tools from Combinatorics, like an argument similar to the Erdős-Szekeres theorem, and a helpful lemma from number theory about 0-sum subsequences.

3.2 Limitation of Local Search

A natural question is whether the approach we presented in this section extends to more graph classes, in particular to all dense graph classes. Here we show that our local search approach fails even for some very dense and very structured classes of graphs. We consider the case of chain graphs.

Recall the definition of a chain graph from Section 2.1. Note that chain graphs can be sparse (e.g. an empty graph is a chain graph), but when required to contain a perfect matching, the graph has to be dense. This can be seen by considering the vertex of highest label on one side and observing that it must be connected to all vertices on the other side. By recursively applying this observation, we can see that the number of edges in the graph must be at least $n^2/8$.

The following is an example of a chain graph that does not satisfy the property PSHORT(t) even for $t = n$. Let $G = (X \,\dot\cup\, Y, E)$ be the chain graph defined by $|X| = |Y| = n$, $N(x_i) := \{y_{n-i+1}, \dots, y_n\}$ for all $1 \leq i \leq |X|$ (see Figure 2). Observe that it is a valid chain graph and that $M := \{\{x_i, y_{n-i+1}\}$, for $1 \leq i \leq |X|\}$ is a perfect matching. Also observe that there is no edge of the form $\{x_i, y_j\}$ for $1 \leq i \leq n-j \leq n$ as required for the property PSHORT(t) (since the graph is bipartite, no edges of the form $\{x_i, x_j\}$ or $\{y_i, y_j\}$ exist either).



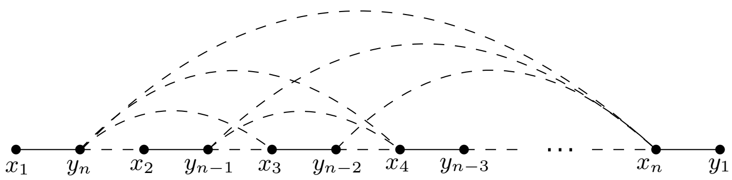


Figure 2 An example of a chain graph G and a perfect matching M in G (left figure), where there exists an M-alternating path with n edges from M (right figure). Edges in M are bold. The property PSHORT(n) is violated on this path.

4 Extending Karzanov's Characterization

In this section, we extend the characterization of exact matchings given by Karzanov [27] for complete and complete bipartite graphs to chain graphs, unit interval graphs, and complete r-partite graphs. Moreover, we exploit this to give deterministic polynomial-time algorithms for Em on those graph classes. This complements the results of Section 3, as chain graphs and unit interval graphs are not captured by the local search approach. On the other hand, complete r-partite graphs actually fit both frameworks. Note that we only provide a coarse outline here and defer many of the proofs and details to the appendix of the full version of this paper.

Given a colored graph (G, c), we denote by $k_{\min}(G)$ the smallest integer k such that there is a PM M in G with $r(M) = k$, and by $k_{\max}(G)$ the largest integer k such that there is a PM M in G with $r(M) = k$. Assuming that G admits at least one PM, both $k_{\min}(G)$ and $k_{\max}(G)$ exist.

Karzanov [27] proved that unless a given colored complete or balanced complete bipartite graph (G, c) has a very specific structure, there must be a PM M in G with $r(M) = k$ for all $k_{\min}(G) \leq k \leq k_{\max}(G)$. Moreover, he also characterized the special cases where this is violated. In particular, even in those special cases the following property still holds. (We use the symbol $\equiv_2$ to denote equivalence modulo 2).

▶ **Definition 5** (Karzanov's Property). *A colored graph (G, c) satisfies Karzanov's property if for any two PMs M and M' with $r(M) \equiv_2 r(M')$ and any integer k with $r(M) \leq k \leq r(M')$ and $k \equiv_2 r(M) \equiv_2 r(M')$, G admits a PM M'' with $r(M'') = k$.*

In other words, if a graph has Karzanov's property, then if we go in steps of two we always find all possible values of red edges between two given PMs of the same parity. We extend this line of work by proving that all colored chain graphs, unit interval graphs, and complete r-partite graphs satisfy Karzanov's Property too. This allows us to decide Em on these graph classes by using an algorithm for Bounded Correct-Parity Perfect Matching (Bcpm) introduced by [13].

Problem Bcpm
Input: Colored graph (G, c), integer $k \geq 0$.
Question: Is there a PM M in G with $r(M) \leq k$ and $r(M) \equiv_2 k$?

We claim that if a graph has Karzanov's property, then Em reduces to Bcpm. Indeed, let $\mathcal{A}$ be an algorithm for Bcpm. Given a colored graph (G, c) and an integer k, we can call $\mathcal{A}$ with input (G, c) and k to check whether there exists a PM M with $r(M) \leq k$ and $r(M) \equiv_2 k$. Next, assume we compute the inverse coloring $\overline{c}$ with $\overline{c}(e) = $ red if $c(e) = $ blue, and $\overline{c}(e) = $ blue if $c(e) = $ red for all $e \in E$. By calling $\mathcal{A}$ with input $(G, \overline{c})$ and $k' = \frac{n}{2} - k$, we can check the existence of a PM M with $r(M) \geq k$ and $r(M) \equiv_2 k$ in (G, c). If we know that Karzanov's property holds in (G, c), these two pieces of information are sufficient to decide Em. (Note that instead of using the algorithm for Bcpm twice, we can first use an algorithm for the simpler problem Cpm which is defined similarly to Bcpm but without the bound on the number of red edges. Depending on the number of red edges in the output matching we then set the Bcpm input appropriately. Cpm has been shown to be solvable in deterministic polynomial time on general graphs [13].)

▶ **Observation 6.** *Em reduces to Bcpm in colored graphs that satisfy Karzanov's property.*

Recall that our goal is to give deterministic polynomial-time algorithms for EM in unit interval graphs, chain graphs, and complete r-partite graphs. Observation 6 provides a possible strategy to achieve this. In particular, we will now proceed to give a condition on graphs that implies Karzanov's property. As it turns out, the same condition is also sufficient to give deterministic polynomial-time algorithms for BCPM.

4.1 A Sufficient Condition for Karzanov's Property

We will now present a sufficient condition for Karzanov's property. As it turns out, the condition is also sufficient to give deterministic polynomial-time algorithms for BCPM. Our condition is based on the existence of certain chord structures in all even-length cycles of a given graph. To state it, we first need to introduce some terminology for chords.

▶ **Definition 7** (Odd Chord, Even Chord, Split of a Chord)**.** *Let C be a cycle in a graph $G = (V, E)$. An edge $e \in E$ is a chord of C if and only if both endpoints of e are on C but $e \notin C$. Let now $e = \{u, v\}$ be a chord of C and consider the paths P_1, P_2 obtained by splitting C at u and v i.e. $C = P_1 \dot\cup P_2$. We call e an odd chord of C if and only if either P_1 or P_2 has odd length. Otherwise, e is called an even chord of C. The split of e is the minimum of the lengths of P_1 and P_2.*

Note that the above definition technically allows C to have even or odd length, but in general we will only be interested in chords of even-length cycles here.

▶ **Definition 8** (Adjacent Chords)**.** *Let C be a cycle in a graph G with chords $e = \{x, y\} \in E$ and $f = \{u, v\} \in E$ whose endpoints appear on C in the order u, v, x, y. Then e and f are said to be adjacent chords of C if additionally, we have $\{v, x\} \in C$ and $\{u, y\} \in C$.*

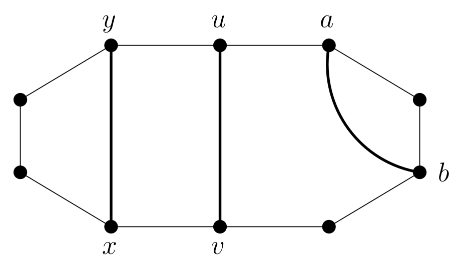


Figure 3 An example of a 10-cycle with three chords. The chords $\{x, y\}$ and $\{u, v\}$ are adjacent and they are both odd chords. Conversely, $\{a, b\}$ is an even chord with split 2. The split of $\{x, y\}$ is 3 and the split of $\{u, v\}$ is 5.

An example of these definitions is found in Figure 3. Given these definitions, we are now ready to state our sufficient condition. Note that the condition is only about the graph structure, i.e. the coloring is irrelevant here.

▶ **Definition 9** (Chord Property)**.** *A simple graph $G = (V, E)$ satisfies the chord property if*

1. *every even cycle C of length at least 6 either has an odd chord or all possible even chords, and*
2. *every even cycle C of length at least 8 either has two adjacent odd chords or all possible even chords with split at least 4.*

As it turns out (see appendix of the full version of this paper), chain graphs, unit interval graphs, and complete r-partite graphs all satisfy the chord property. In fact, the even chords are only needed in complete r-partite graphs. In other words, chain graphs and unit interval graphs satisfy the chord property without making use of the parts about even chords. By our next lemma, this means that all three graph classes satisfy Karzanov's property for every possible coloring.

▶ **Lemma 10** (Chord Property is Sufficient). *Let G be an arbitrary graph satisfying the chord property and let c be an arbitrary coloring of G. Then the colored graph (G, c) satisfies Karzanov's property.*

Proof. Deferred to the appendix of the full version of this paper. ◀

Note that the reverse is not true, i.e. given a colored graph (G, c) with Karzanov's property, it is not necessarily the case that G has the chord property.

In order to solve EM on graphs satisfying the chord property, it remains to give a deterministic polynomial-time algorithm for BCPM on those graphs.

▶ **Lemma 11** (BCPM on Graphs with the Chord Property). *There is a deterministic polynomial-time algorithm that decides BCPM correctly for all inputs where the graph satisfies the chord property.*

Proof. Deferred to the appendix of the full version of this paper. ◀

Finally, we can combine Observation 6 with Lemma 10 and Lemma 11 to get the main result of this section.

▶ **Theorem 12.** *There is a deterministic polynomial-time algorithm that decides EM on all colored graphs (G, c) where G satisfies the chord property.*

Proof. The colored graph (G, c) satisfies Karzanov's property by Lemma 10. By Observation 6, this reduces deciding EM for (G, c) to deciding BCPM. Moreover, the graph G remains unaltered in this reduction. Hence, this can be achieved in deterministic polynomial-time with the algorithm from Lemma 11. ◀

We prove in the appendix of the full version of this paper that unit interval graphs, chain graphs, and complete r-partite graphs satisfy the chord property. We conclude that EM restricted to those graph classes can be decided in deterministic polynomial-time.

4.2 Limitation of Karzanov's Property

A natural question is whether the approach we presented in this section extends to more graph classes. In particular, interval graphs and bipartite interval graphs would be suitable candidates as they are superclasses of unit interval graphs and chain graphs, respectively.

Unfortunately, it turns out that Karzanov's Property is violated on both graph classes. Concrete counterexamples are given in Figures 4 and 5.

While our approach fails to generalize to these graph classes, there still seems to be some hope. Consider the following property of colored graphs, which we coined *Karzanov's weak property.*

▶ **Definition 13** (Karzanov's Weak Property). *A colored graph (G, c) satisfies Karzanov's weak property if for any two PMs M and M' and integer k with $r(M) \leq k \leq r(M')$, there is a PM M'' with $r(M'') \in \{k, k + 1\}$.*

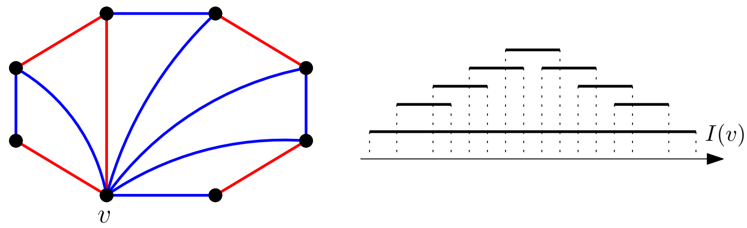


Figure 4 On the left we have a colored interval graph on eight vertices which does not satisfy Karzanov's property. In particular, there are PMs with 0, 1, 3, and 4 red edges but there is no PM with exactly 2 red edges. The interval representation of the graph is given on the right. Interval $I(v)$ corresponds to vertex v from the left. Note that the vertical position of the intervals is irrelevant, only the relative horizontal position of the intervals matters.

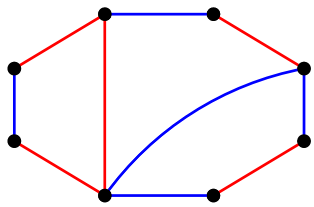


Figure 5 By deleting the even chords from the graph in Figure 4, we obtain a bipartite interval graph. It admits the same PMs as the interval graph in Figure 4. Hence, this colored graph also violates Karzanov's property.

The main difference to Karzanov's property is that we are missing the constraint on the parity of the number of red edges. Consider e.g. a graph with exactly two PMs with 0 and 3 red edges, respectively. Such a graph would satisfy Karzanov's property but violate Karzanov's weak property. In particular, Karzanov's property does not imply Karzanov's weak property. Still, compared to Karzanov's property, Karzanov's weak property gives us less structure to work with and typically holds on larger graph classes. Unfortunately, we have not yet been able to solve EM using Karzanov's weak property.

As it turns out, all colored bipartite chordal and strongly chordal graphs satisfy Karzanov's weak property.

▶ **Lemma 14.** *All colored bipartite chordal and strongly chordal graphs satisfy Karzanov's weak property.*

Proof. Deferred to the appendix of the full version of this paper. ◀

In particular, the same observation holds in all colored interval and bipartite interval graphs as they are subclasses of strongly chordal and bipartite chordal graphs, respectively.

5 Conclusion

In this paper we made substantial progress towards solving the notoriously difficult Exact Matching problem, in particular in the regime of dense graphs. We provide general frameworks that not only encompass all previously known results for these types of graphs, but also include a multitude of graph classes for which the problem is now solved. We remark that it is inherent to our techniques that they will fail on sparse graphs: It seems very unlikely that a local search on a sparse graph is successful (since changing one edge of a PM in a sparse graph often times requires changing many more edges). It is also unlikely that a sparse graph has Karzanov's property: On sparse graphs, we do not expect the solution landscape to be dense. Still, we hope that our approach sheds further light onto these questions. One could imagine, for example, to split a graph into a dense and a sparse part, and apply different techniques to different parts.

In this paper, we also provided some open questions that are reasonable to attack next, since they seem to be in reach of current methods. In particular, is it possible to have a deterministic poly-time algorithm for $G(n, 1/2)$? Can one find a deterministic poly-time algorithm for those graph classes where the weak Karzanov property holds? (Interval graphs, bipartite interval graphs, strongly chordal graphs, bipartite chordal graphs.) Note that for graphs with the weak Karzanov property, we can always find a PM with either $k-1$ or k red edges, but the final decision if k can be achieved still seems difficult.

References

1 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004.

2 Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

3 Stephan Artmann, Robert Weismantel, and Rico Zenklusen. A strongly polynomial algorithm for bimodular integer linear programming. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1206–1219, 2017.

4 Vikraman Arvind, Johannes Köbler, Sebastian Kuhnert, and Jacobo Torán. Solving linear equations parameterized by hamming weight. *Algorithmica*, 75(2):322–338, 2016.

5 André Berger, Vincenzo Bonifaci, Fabrizio Grandoni, and Guido Schäfer. Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Mathematical Programming*, 128(1):355–372, 2011.

6 Daniel P. Bovet and Pierluigi Crescenzi. *Introduction to the Theory of Complexity*. Prentice Hall International (UK) Ltd., GBR, 1994.

7 Paolo M. Camerini, Giulia Galbiati, and Francesco Maffioli. Random pseudo-polynomial algorithms for exact matroid problems. *Journal of Algorithms*, 13(2):258–273, 1992.

8 Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An EPTAS for Budgeted Matching and Budgeted Matroid Intersection via Representative Sets. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

9 Anita Dürr, Nicolas El Maalouly, and Lasse Wulf. An approximation algorithm for the exact matching problem in bipartite graphs. *arXiv preprint arXiv:2307.02205*, 2023.

10 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965.

11 Nicolas El Maalouly. Exact matching: Algorithms and related problems. *arXiv preprint arXiv:2203.13899*, 2022.

12 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)*, volume 241 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 46:1–46:14, 2022.

13 Nicolas El Maalouly, Raphael Steiner, and Lasse Wulf. Exact matching: Correct parity and FPT parameterized by independence number. *CoRR*, abs/2207.09797, 2022. doi:10.48550/arXiv.2207.09797.

14 Nicolas El Maalouly and Yanheng Wang. Counting perfect matchings in dense graphs is hard. *arXiv preprint arXiv:2210.15014*, 2022.

15 Nicolas El Maalouly and Lasse Wulf. Exact matching and the top-k perfect matching problem. *arXiv preprint arXiv:2209.09661*, 2022.

16 Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.

17 Dennis Fischer, Tim A Hartmann, Stefan Lendl, and Gerhard J Woeginger. An investigation of the recoverable robust assignment problem. *arXiv preprint arXiv:2010.11456*, 2020.

18 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.

19 Hans-Florian Geerdes and Jácint Szabó. A unified proof for Karzanov's exact matching theorem. Technical Report QP-2011-02, Egerváry Research Group, Budapest, 2011.

20 Martin Charles Golumbic. *Algorithmic graph theory and perfect graphs.* Elsevier, 2004.

21 Fabrizio Grandoni and Rico Zenklusen. Optimization with more than one budget. *arXiv preprint arXiv:1002.2147*, 2010.

22 Rohit Gurjar, Arpita Korwar, Jochen Messner, Simon Straub, and Thomas Thierauf. Planarizing gadgets for perfect matching do not exist. In *International Symposium on Mathematical Foundations of Computer Science*, pages 478–490. Springer, 2012.

23 Rohit Gurjar, Arpita Korwar, Jochen Messner, and Thomas Thierauf. Exact perfect matching in complete graphs. *ACM Transactions on Computation Theory (TOCT)*, 9(2):1–20, 2017.

24 Russell Impagliazzo and Avi Wigderson. P= BPP if e requires exponential circuits: Derandomizing the xor lemma. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 220–229, 1997.

25 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)*, pages 1635–1654. SIAM, 2023.

26 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 355–364, 2003.

27 AV Karzanov. Maximum matching of given weight in complete and complete bipartite graphs. *Cybernetics*, 23(1):8–13, 1987.

28 Steven Kelk and Georgios Stamoulis. Integrality gaps for colorful matchings. *Discrete Optimization*, 32:73–92, 2019.

29 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64:19–37, 2012.

30 László Lovász. Matching structure and the matching lattice. *Journal of Combinatorial Theory, Series B*, 43:187–222, 1987.

31 Monaldo Mastrolilli and Georgios Stamoulis. Constrained matching problems in bipartite graphs. In *International Symposium on Combinatorial Optimization*, pages 344–355. Springer, 2012.

32 Monaldo Mastrolilli and Georgios Stamoulis. Bi-criteria and approximation algorithms for restricted matchings. *Theoretical Computer Science*, 540:115–132, 2014.

33 Ketan Mulmuley, Umesh V Vazirani, and Vijay V Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

34 Martin Nägele, Christian Nöbel, Richard Santiago, and Rico Zenklusen. Advances on strictly δ-modular ips. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 393–407. Springer, 2023.

35 Yoshio Okamoto, Ryuhei Uehara, and Takeaki Uno. Counting the number of matchings in chordal and chordal bipartite graph classes. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 296–307. Springer, 2009.

36 Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM (JACM)*, 29(2):285–309, 1982.

37 Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980.

38 Georgios Stamoulis. Approximation algorithms for bounded color matchings via convex decompositions. In *International Symposium on Mathematical Foundations of Computer Science*, pages 625–636. Springer, 2014.

39 Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-NC. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 696–707. Ieee, 2017.

40 Moshe Y Vardi and Zhiwei Zhang. Quantum-inspired perfect matching under vertex-color constraints. *arXiv preprint arXiv:2209.13063*, 2022.

41 Vijay V Vazirani. NC algorithms for computing the number of perfect matchings in k3, 3-free graphs and related problems. *Information and computation*, 80(2):152–164, 1989.

42 Tongnyoul Yi, Katta G Murty, and Cosimo Spera. Matchings in colored bipartite networks. *Discrete Applied Mathematics*, 121(1-3):261–277, 2002.

43 Raphael Yuster. Almost exact matchings. *Algorithmica*, 63(1):39–50, 2012.

44 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International symposium on symbolic and algebraic manipulation*, pages 216–226. Springer, 1979.