Parameterized Complexity of Perfectly Matched Sets

Akanksha Agrawal
Indian Institute of Technology Madras, Chennai, India

Sutanay Bhattacharjee
Indian Institute of Technology Madras, Chennai, India

Satyabrata Jana
The Institute of Mathematical Sciences, HBNI, Chennai, India

Abhishek Sahu
Indian Institute of Technology Madras, Chennai, India

Abstract

For an undirected graph G, a pair of vertex disjoint subsets (A, B) is a *pair of perfectly matched sets* if each vertex in A (resp. B) has exactly one neighbor in B (resp. A). In the above, the size of the pair is $|A|$ $(= |B|)$. Given a graph G and a positive integer k, the Perfectly Matched Sets problem asks whether there exists a pair of perfectly matched sets of size at least k in G. This problem is known to be NP-hard on planar graphs and W[1]-hard on general graphs, when parameterized by k. However, little is known about the parameterized complexity of the problem in restricted graph classes. In this work, we study the problem parameterized by k, and design FPT algorithms for: i) apex-minor-free graphs running in time $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$, and ii) $K_{b,b}$-free graphs. We obtain a linear kernel for planar graphs and $k^{\mathcal{O}(d)}$-sized kernel for d-degenerate graphs. It is known that the problem is W[1]-hard on chordal graphs, in fact on split graphs, parameterized by k. We complement this hardness result by designing a polynomial-time algorithm for interval graphs.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases Perfectly Matched Sets, Parameterized Complexity, Apex-minor-free graphs, d-degenerate graphs, Planar graphs, Interval Graphs

Digital Object Identifier 10.4230/LIPIcs.IPEC.2022.2

Funding *Akanksha Agrawal*: Supported by New Faculty Initiation Grant no. NFIG008972.

1 Introduction

Matching is one of the very classical polynomial-time solvable problems in Computer Science with varied applications. Finding a matching with additional structure, such as an induced matching has been well studied both in classical complexity as well as parameterized complexity, see, for instance, [4, 9, 18, 20, 24, 24, 27, 28] (list is only illustrative, and not comprehensive). In this article, we are interested in a matching that is slightly weaker than the structure of an induced matching but still more structured than a matching.

For a graph G, a pair of vertex disjoint subsets, (A, B) is a *pair of perfectly matched sets* in G if each vertex in A has exactly one neighbor in B and each vertex in B has exactly one neighbor in A; the *size* of the pair is $|A|$ $(= |B|)$. Note that there can be edges between vertices of A (resp. B), which is forbidden in the case of induced matching. We study the problem called Perfectly Matched Sets, which is defined below.

Perfectly Matched Sets	**Parameter:** k
Input: An undirected graph G and an integer k.	
Question: Does there exist a pair of perfectly matched sets of size at least k in G?	

17th International Symposium on Parameterized and Exact Computation (IPEC 2022).
Editors: Holger Dell and Jesper Nederlof; Article No. 2; pp. 2:1–2:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This problem was first introduced in [27] where it was named as *Maximum TR-matching problem* (Transmitter- Receiver problem). The paper showed that this problem is NP-complete when restricted to graphs having degree 3. Evan, Goldreich, and Tong in [13] showed that TR-matching is NP-complete on bipartite graphs. This problem was revisited by Aravind and Saxena in 2021, [1] where they called the problem as PERFECTLY MATCHED SETS. They designed FPT algorithms for this problem parameterized by the structural parameters such as distance to cluster, distance to co-cluster, and treewidth. They also prove that the problem is NP-hard on planar graphs and W[1]-hard parameterized by the solution size k, when restricted to bipartite graphs and split graphs.

The PERFECTLY MATCHED SETS problem is also closely related to the problem PERFECT MATCHING CUT where we want edge cuts of size k, such that the vertices participating in these edges induce a matching and a perfect matching, respectively. We remark that in PERFECTLY MATCHED SETS, we do not insist that the edges between the pair of perfectly matched sets (A, B) is a cut in the graph. The MATCHING CUT and PERFECT MATCHING CUT problems have been investigated in the literature even when restricted to well-studied graph classes, see, for instance, [2, 6, 7, 21, 22, 25].

Our Results. In this paper, we investigate the parameterized complexity of the PERFECTLY MATCHED SETS problem when the input graph is from a structured graph family, for several choices of well-studied graph families. The starting point of our work is the result by Aravind and Saxena [1]. The paper showed that the problem is W[1]-hard even on split graphs, which is an important subclass of chordal graphs. Inspired by this negative result, we turn to interval graphs, which is arguably the most well-studied subclass of chordal graphs. We obtain the following result by using a dynamic programming based algorithm.

▶ **Theorem 1.** PERFECTLY MATCHED SETS *on interval graphs admits an algorithm running in time $\mathcal{O}(n^5)$.*

Aravind and Saxena [1] showed that PERFECTLY MATCHED SETS is NP-complete even when the input graph is planar. Inspired by this we design an FPTalgorithm for a strictly more general class of apex-minor-free graphs. A graph H is an *apex* graph if there is $v \in V(H)$, such that $H - \{v\}$ is planar. Consider any finite set $\mathcal{H}$ of graphs that contains at least one apex graph, and let $\mathcal{F}_{\mathcal{H}}$ be the family of graphs that do not contain any graph from $\mathcal{H}$ as a minor. The $\mathcal{H}$-MINOR FREE PMS problem is the PERFECTLY MATCHED SETS problem with an additional guarantee that the input graph belongs to $\mathcal{F}_{\mathcal{H}}$. Note that for $\mathcal{H} = \{K_5, K_{3,3}\}$, $\mathcal{F}_{\mathcal{H}}$ is the family of planar graphs. We obtain the following result:

▶ **Theorem 2.** *For any (fixed) finite set $\mathcal{H}$ of graphs that contains at least one apex graph,* $\mathcal{H}$-MINOR FREE PMS *has an FPT algorithm running in time $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$.*

We remark that the same approach used in obtaining the above result can be used to obtain an FPT algorithm on bounded genus graphs, due to bidimensionality [10]. We remark that having a pair of perfectly matched sets of size at least k is expressible in MSO (actually, even in FO). So, there is an FPT algorithm on the much more general nowhere dense classes (admittedly with a worse running time)[16].

For $b \in \mathbb{N}$, a graph is $K_{b,b}$-free if it does not contain a bi-clique with b vertices on each side as a subgraph. We obtain the following result by using an approach similar to random separation [3], in combination with a result of Dabrowski et al. [9].

▶ **Theorem 3.** *For any fixed $b \in \mathbb{N}$,* PERFECTLY MATCHED SETS *on $K_{b,b}$-free graphs admits an FPT algorithm, when parameterized by k.*

Kanj et al. [18] and Erman et al. [20] independently designed $\mathcal{O}(k^c)$ kernels for the Induced Matching problem for graphs of arboricity bounded by c . The authors [18] also showed that any twinless graph of average degree d and bounded chromatic number contains an induced matching of size $\Omega(n^{1/d})$. The core of their proof is the *system of strong representatives* of a set family. This combinatorial tool also forms the backbone of our following result.

▶ **Theorem 4.** Perfectly Matched Sets *admits a $k^{\mathcal{O}(d)}$-sized kernel on d-degenerate graphs.*

As planar graphs are 5-degenerate, the theorem above directly gives us a polynomial kernel for Perfectly Matched Sets on these graphs. Following an approach by Kanj et al. [18] for obtaining a linear kernel for Induced Matching on planar graphs, we obtain a linear kernel (improving upon the already obtained polynomial kernel) for Perfectly Matched Sets on this graph class.

2 Preliminaries

Sets and graph notations. We use $\mathbb{N} = \{1, 2, \ldots\}$ to denote the set of natural numbers. We use $[k]$ as a shorthand for $\{1, 2, \ldots, k\}$ and use $[k]_0$ for $[k] \cup \{0\}$, where $k \in \mathbb{N}$. In this article, we only consider simple undirected graphs. Given a graph G, we denote the vertex set and edge set of G by $V(G)$ and $E(G)$ respectively. Unless specified, n and m denote the number of vertices and edges of the graph G. Two vertices u, v are said to be *adjacent* if there is an edge (denoted by $\{u, v\}$) between u and v in G. For $X \subseteq V(G)$, $G[X]$ denotes the induced subgraph of G with vertex set X and edge set $\{\{u, v\} \mid u, v \in X \text{ and } \{u, v\} \in E(G)\}$, $G - X$ denotes the subgraph $G[V(G)\backslash X]$. For an edge set $E' \subseteq E$, $V(E')$ denotes the set of all the vertices of G having at least one edge in E' incident on it. $E(A, B)$ denotes the set of edges with one endpoint in A and the other in B. The open neighborhood of a vertex v, denoted by $N_G(v)$, is the set of vertices adjacent to v. The *closed neighborhood* of v is defined as $N_G[v] = N_G(v) \cup \{v\}$. The subscript in the notation for neighborhood is omitted if the graph under consideration is clear. For $X \subseteq V(G)$, $N[X]$ denotes the set of vertices $\bigcup_{v \in X} N[v]$. Two distinct vertices u, v is said to be a pair of *false twins* if $N_G(u) = N_G(v)$ and *true twins* if $N_G[u] = N_G[v]$. A *clique* in graph G is a set of vertices such that there is an edge between every pair of vertices in the set. An *independent set* in the graph G is a set of vertices such that there is no edge between any pair of vertices in the set. $K_{n,m}$ is the complete bipartite graph, also known as a biclique, with partitions of size n and m. A k-biclique is a $2k$-vertex complete bipartite graph. A subset $D \subseteq V(G)$ is said to be a dominating set of G if $N[D] = V(G)$. A *vertex cover* of a graph is a set of vertices that includes at least one endpoint of every edge of the graph. The cardinality of the smallest size dominating set is called as *domination number* of G. D is said to be a *2-dominating set* if $N[N[D]] = V(G)$. $G\backslash e$ denotes the graph obtained by contracting the edge e in G. The contraction of an edge $\{u, v\}$ in the graph involves the deletion of vertices u and v from G and the addition of a new vertex w, which is adjacent to all the vertices of $N(u) \cup N(v)$. For two graphs G_1 and G_2, we denote $G_1 \subseteq G_2$ if G_1 is an induced subgraph of G_2.

Graph classes. A graph is *planar* if it can be drawn in the plane without edge intersections except at the endpoint). A graph G is a *d-degenerate* graph if every induced subgraph of G contains a vertex of degree at most d. A $K_{b,b}$-free graph is a graph that does not contain biclique $K_{b,b}$ as a subgraph (not necessarily induced). An *apex graph* is a graph that can be made planar by removing one of its vertices. Apex-minor-free graphs are basically those graphs that exclude a fixed apex graph as a minor. More precisely, $\mathcal{C}$ is apex-minor-free graph class if there exists some apex graph H such that no graph from $\mathcal{C}$ admits H as a minor. An *interval graph* is an undirected graph formed from a set of intervals on the real line, with a vertex for each interval and an edge between vertices whose intervals intersect. It is the intersection graph of the intervals. [5]. For standard graph definition and notations, we refer to the graph theory book by R. Diestel [11]. For parameterized complexity terminology, we refer to the parameterized algorithms book by Cygan et al. [8].

Treewidth. A *tree decomposition* of a graph $G = (V, E)$ is a pair (T, X) where T is a tree on vertex set $V(T)$. The vertices of $V(T)$ are called *nodes*. Also, $X = (\{X_i | \, i \in V(T)\})$ is a collection of subsets of V such that -

1. Every vertex of G is contained in at least one bag. $\cup_{i \in V(T)} X_i = V$,
2. For every edge $\{u, v\} \in E$, there exists a node $i \in V(T)$ such that bag X_i contains both u and v.
3. For each $u \in V$, the set of nodes whose bags contain u, $T_u = \{i \in V(T) : i \in X_i\}$ forms a connected subtree of T.

The *width* of a tree decomposition $(T, (\{X_i | \, i \in V(T)\})$ is equal to the maximum size of its bag minus 1, $max_{i \in V(T)}\{|X_i| - 1\}$. The *treewidth* of a graph G, $\text{tw}(G)$ is the minimum width of a tree decomposition over all tree decompositions of G.

Perfectly matched sets. A *matching* in a graph G is a set of edges M such that no two edges in M share the same endpoint. A matching M is *maximal* if $G - V(M)$ is edge less. A matching M is said to be an *induced matching* if the subgraph induced by the vertices in M contains only the edges of M. If M is maximal then $V(M)$ is a vertex cover of G, and it is easy to verify that $\text{tw}(G) \leqslant |V(M)|$. For a pair (A, B) of disjoint subsets of vertices of $V(G)$, we say (A, B) is a *pair of perfectly matched sets* if every vertex in A (resp. B) has exactly one neighbor in B (resp. A). The size of the pair is $|A| = |B|$.

Parameterized problems and kernels. A parameterized problem Π is a subset of $\Gamma^* \times \mathbb{N}$ for some finite alphabet Γ. An instance of a parameterized problem consists of (X, k), where k is called the parameter. The notion of kernelization is formally defined as follows. A kernelization algorithm, or in short, a kernelization, for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}$ is an algorithm that, given $(X, k) \in \Gamma^* \times \mathbb{N}$, outputs in time polynomial in $|X| + k$ a pair $(X', k') \in \Gamma^* \times \mathbb{N}$ such that (a) $(X, k) \in \Pi$ if and only if $(X', k') \in \Pi$ and (b) $|x'|, |k| \leqslant g(k)$, where g is some computable function depending only on k. The output of kernelization (X', k') is referred to as the kernel and the function g is referred to as the size of the kernel. If $g(k) \in k^{\mathcal{O}(1)}$, then we say that Π admits a polynomial kernel. We refer to the monographs [12, 14, 26] for a detailed study of the area of kernelization.

3 Polynomial-time Algorithm for Interval Graphs

Recall that Perfectly Matched Sets is W[1]-hard when parameterized by the solution size k even when restricted to split graphs (and thus, chordal graphs). Interval graphs belong to the class of chordal graphs. In this section, we present a polynomial-time dynamic programming algorithm that computes a maximum-sized pair of perfectly matched sets for any given interval graph.

Let G be an interval graph with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$. Since G is an interval graph, there exists a corresponding geometric intersection representation of G, where each vertex $v_i \in V(G)$ is associated with an interval $I_i = [\ell(I_i), r(I_i)]$ in the real line, where $\ell(I_i)$ and $r(I_i)$ denote left and right endpoints, respectively in I_i. Two vertices v_i and v_j are adjacent in G if and only if their corresponding intervals I_i and I_j intersect with each other. We can also assume that along with the graph, we are also given the corresponding underlying intervals on the real line, as there are well-known linear-time algorithms that compute such a representation [19]. We use $\mathcal{I}$ to denote the set $\{I_i \colon v_i \in V\}$ of intervals and P to denote the set of all endpoints of these intervals, i.e., $P = \cup_{I \in \mathcal{I}} \{\ell(I), r(I)\}$. In the remaining section, we will use v_i and I_i interchangeably. Note that we can assume that the endpoints of all the intervals in the interval representation are distinct − otherwise, we can slightly perturb the endpoints of the intervals to obtain a new interval representation of the graph in which this is true.

▶ **Proposition 5.** *Let G be a connected interval graph. There exists an ordering, $\prec$, of $V(G)$ such that for $u, v, w \in V(G)$ if $u \prec v \prec w$ and $\{u, w\} \in E(G)$ then $\{v, w\} \in E(G)$.*

We remark that such an ordering in Proposition 5 can be obtained based on the right endpoints of intervals, more specifically the set $\{r(I_i)\}$ and the ordering is as follows: for any two vertices v_i and v_j, we have $v_i \prec v_j$ if and only if $r(I_i) < r(I_j)$. We call such an ordering, the *right-end ordering* of $V(G)$.

▶ **Lemma 6.** *Let G be an interval graph with a right-end ordering, $\prec$, of $V(G)$. Consider any distinct pair of edges $\{u, v\}$ and $\{u', v'\}$ in a pair of perfectly matched sets (A, B) where $u \prec v$ and $u' \prec v'$. If $u \prec u'$, then $v \prec u'$.*

Proof. Towards a contradiction suppose there are edges $\{u, v\}, \{u', v'\}$ in the pair of perfectly matched sets (A, B), where $u \prec v$, $u' \prec v'$, $u \prec u'$ and $u' \prec v$. Then, either $u \prec u' \prec v' \prec v$, or $u \prec u' \prec v \prec v'$. In either of these cases, by Proposition 5, v is adjacent to both u' and v' which is a contradiction to the fact that (A, B) is perfectly matched sets in G. ◀

Lemma 6 directly implies the following remark.

▶ Remark 7. Let $\{\{u_i, v_i\} \colon 1 \leqslant i \leqslant k\}$ be a set of k edges in a pair (A, B) of perfectly matched sets in G with $u_1 \prec u_2 \prec \ldots \prec u_k$ and $u_i \prec v_i$, for each $i \in [k]$. Then, $u_1 \prec v_1 \prec u_2 \prec v_2 \prec \ldots \prec u_k \prec v_k$.

Algorithm and its Correctness. We define a table for our dynamic-programming algorithm. Let $v_1 \prec v_2 \prec \ldots \prec v_n$ be the right-end ordering of the vertex set of G. For every tuple (v_i, v_j, t), where $\{v_i, v_j\} \in E(G)$, $i, j \in [n]$, $i < j$ and $t \in [\lfloor n/2 \rfloor]$, we define two Boolean values: (i) $\texttt{PM}\big[(v_i, A), (v_j, B); t\big]$ and (ii) $\texttt{PM}\big[(v_i, B), (v_j, A); t\big]$.[1] The entry $\texttt{PM}\big[(v_i, A), (v_j, B); t\big]$ is `true` if there exists a pair (A, B) of perfectly matched sets of size t such that $v_i \in A$, $v_j \in B$ and for all the vertices $v \in (A \cup B) \setminus \{v_i, v_j\}$, we have $v \prec v_i$. Similarly the entry $\texttt{PM}\big[(v_i, B), (v_j, A); t\big]$ is `true` if there exists a pair (A, B) of perfectly matched sets of size t such that $v_i \in B$, $v_j \in A$ and for all the vertices $u \in (A \cup B) \setminus \{v_i, v_j\}$, we have $u \prec v_i$.

In the base case, both $\texttt{PM}\big[(v_i, A), (v_j, B); 1\big]$ and $\texttt{PM}\big[(v_i, B), (v_j, A); 1\big]$ are `true` for every possible pair v_i and v_j (note because of the way the entry is defined, $\{v_i, v_j\}$ must be an edge in G). We will use the convention that *empty* OR is 0. In the lemma below, we give a recursive formula for computing the values $\texttt{PM}\big[(v_i, A), (v_j, B); t\big]$ for $t > 1$.

[1] A and B in these entries are just symbols, added for extra clarity.

▶ **Lemma 8.** *For every integer $t \in [\lfloor n/2 \rfloor] \backslash \{1\}$, and every pair of adjacent vertices v_i, v_j in G where $i < j$, the following recurrence holds:*

$$\texttt{PM}\big[(v_i, A), (v_j, B); t\big] = \bigvee_{\substack{\{x,y\} \in E(G) \\ x \prec y \prec v_i}} \Big(\Big(\texttt{PM}\big[(x, A), (y, B); t-1\big] \wedge [\{x, v_j\} \notin E(G)] \wedge [\{y, v_i\} \notin E(G)] \Big) \bigvee \Big(\texttt{PM}\big[(x, B), (y, A); t-1\big] \wedge [\{x, v_i\} \notin E(G)] \wedge [\{y, v_j\} \notin E(G)] \Big) \Big)$$

Proof. In the forward direction let us assume that $\texttt{PM}\big[(v_i, A), (v_j, B); t\big] = \texttt{true}$. So according to the definition of our dynamic-programming table, $\{v_i, v_j\} \in E(G)$ and there exists a pair (A, B) of perfectly matched sets of size t such that $v_i \in A$, $v_j \in B$ and for all the vertices $v \in (A \cup B) \backslash \{v_i, v_j\}$, we have $v \prec v_i$. Now consider the pair $(A' = A \backslash \{v_i\}, B' = B \backslash \{v_j\})$. It is easy to see that this pair is a perfectly matched sets of size $t - 1$ and all the vertices v in the pair having the property that $v \prec v_i$. Consider the last vertex in the right-end ordering of $V(G)$ which occures in the vertex set $A' \cup B'$. Let this vertex be y and x be its (only) neighbour in B'. Note that $x \prec y$ and for any vertex $v \in (A' \cup B') \backslash \{x, y\}$, it must hold that $v \prec x$ (see Remark 7). If $y \in B'$, then clearly, $\{x, v_j\} \notin E(G)$, $\{y, v_i\} \notin E(G)$, and $\texttt{PM}\big[(x, A), (y, B); (t-1)\big] = \texttt{true}$. Otherwise, $y \in A'$, and then $\{x, v_i\} \notin E(G)$, $\{y, v_j\} \notin E(G)$, and $\texttt{PM}\big[(x, B), (y, A); (t-1)\big] = \texttt{true}$.

In the reverse direction, assume that there exists a pair of vertices $x \prec y$, $\{x, y\} \in E(G)$ such that $\texttt{PM}\big[(x, A), (y, B); (t-1)\big] = \texttt{true}$ and $\{x, v_j\} \notin E(G)$, $\{y, v_i\} \notin E(G)$. (The case when $\texttt{PM}\big[(x, B), (y, A); t-1\big] = \texttt{true}$ and $\{x, v_i\} \notin E(G)$, $\{y, v_j\} \notin E(G)$ can be argued symmetrically.) The above means that there is a pair of perfectly matched sets (A', B') with $t-1$ edges such that: $\{x, y\} \in E(G)$, $x \in A'$, $y \in B'$, and for each $v \in (A' \cup B') \backslash \{x, y\}$, we have $v \prec x$. Let $A = A' \cup \{v_i\}$ and $B = B' \cup \{v_j\}$. Note that we have $x \prec y \prec v_i \prec v_j$, and thus, for each $v \in A' \cup B'$, we have $v \prec v_i \prec v_j$. For a contradiction suppose that we have $v \in B'$, such that $\{v, v_i\} \in E(G)$. Note that $v \prec x \prec y \prec v_i$, as $\{y, v_j\} \notin E(G)$ (see Remark 7). But then from Lemma 6, we can obtain that $\{v, x\} \in E(G)$, which contradicts that (A', B') is a pair of perfectly matched sets. Similarly, towards a contradiction suppose that we have $v \in A'$, such that $\{v, v_j\} \in E(G)$. Then, $v \prec x \prec y \prec v_j$, and thus, $\{y, v\} \in E(G)$, which is a contradiction. From the above discussions, we can conclude that $\texttt{PM}\big[(v_i, A), (v_j, B); t\big] = \texttt{true}$. ◀

Similarly, we have a recursive formula for computing the values $\texttt{PM}\big[(v_i, B), (v_j, A); t\big]$ for $t > 1$. The correctness proof is similar to that of Lemma 8.

▶ **Lemma 9.** *For every integer $t \in [\lfloor n/2 \rfloor] \backslash \{1\}$, and every pair of adjacent vertices v_i, v_j in G where $i < j$, the following recurrence holds:*

$$\texttt{PM}\big[(v_i, B), (v_j, A); t\big] = \bigvee_{\substack{\{x,y\} \in E(G) \\ x \prec y \prec v_i}} \Big(\Big(\texttt{PM}\big[(x, A), (y, B); t-1\big] \wedge [\{y, v_j\} \notin E(G)] \wedge [\{x, v_i\} \notin E(G)] \Big) \bigvee \Big(\texttt{PM}\big[(x, B), (y, A); t-1\big] \wedge [\{x, v_j\} \notin E(G)] \wedge [\{y, v_i\} \notin E(G)] \Big) \Big)$$

We can compute all the entries of our dynamic programming table using the recurrence relations given by Lemma 8 and Lemma 9.

Time Complexity. For a pair of adjacent vertices v_i, v_j, where $i < j$, the time required to compute $\texttt{PM}\big[(v_i, A), (v_j, B); t\big]$ and $\texttt{PM}\big[(v_i, B), (v_j, A); t\big]$, once we have computed the entries till the values at most $t - 1$, is bounded by $\mathcal{O}(n^2)$. As $t < n$, the number of entries we have to compute is bounded by $\mathcal{O}(n^3)$, thus bounding the total running time of our algorithm by $\mathcal{O}(n^5)$. This proves Theorem 1.

4 FPT Algorithm for Apex-Minor-Free Graphs

Consider any (fixed) finite set $\mathcal{H}$ of graphs that contains at least one apex graph; we will work with this fixed family throughout this section. Recall that $\mathcal{F}_{\mathcal{H}}$ is the family of graphs that do not contain any graph from $\mathcal{H}$ as a minor, and the $\mathcal{H}$-Minor Free PMS problem is the same as the Perfectly Matched Sets problem with an additional guarantee that the input graph belongs to $\mathcal{F}_{\mathcal{H}}$. In this section, we prove Theorem 2 by designing a simple FPT algorithm with the desired running time. Let (G,k) be an instance of $\mathcal{H}$-Minor Free PMS. Our algorithm will begin by greedily trying to construct a solution, if we succeed then the algorithm halts. Otherwise, we will be able to bound the size of a 2-dominating in G by $\mathcal{O}(k)$. This together with a result of Fomin [15]) will imply that the treewidth of G is bounded by $\mathcal{O}(\sqrt{k})$. Now we can use the algorithm of Aravind and Saxena [1] for Perfectly Matched Sets parameterized by treewidth to obtain the proof of the theorem. We begin by stating the two useful results.

▶ **Proposition 10** (Lemma 2, [15]). *For an $\mathcal{H}$-minor free graph G, if ℓ is the size of a minimum 2-dominating set of G, then the treewidth of G is bounded by $c_{\mathcal{H}} \cdot \sqrt{\ell}$, where $c_{\mathcal{H}}$ is a constant depending on $\mathcal{H}$.*

▶ **Proposition 11** (Theorem 7, [1]). *There exists an algorithm that calculate maximum perfectly matched sets for an n vertex graph with treewidth at most w in time $\mathcal{O}(12^w \cdot poly(n))$.*

The next lemma gives the procedure that either resolves the instance or obtains a small 2-dominating set in G.

▶ **Lemma 12.** *There is a polynomial time algorithm that either correctly concludes that (G,k) is a yes-instance of $\mathcal{H}$-Minor Free PMS, or outputs a 2-dominating set Q of G where $|Q| \leqslant 2 \cdot (k-1)$.*

Proof. Let (G,k) be an instance of the problem. If G has an isolated vertex, then such a vertex is not part of any perfectly matched set, and thus we remove it. We will next create a sequence of perfectly matched sets $S_0 \subset S_1 \subset \cdots \subset S_q$ and graphs $G_0 \supseteq G_1 \supseteq \cdots \supseteq G_q$, which, intuitively speaking, will be constructed by greedily adding an edge (one at a time) to form a perfectly matched set.

Initialize $S_0 = \varnothing$ and $G_0 = G$. Iteratively do the following: if there is an edge $e_i = \{u_i, v_i\} \in E(G_i)$, then set $S_{i+1} = S_i \cup \{e\}$ and $G_{i+1} = G_i - (N_G[u] \cup N_G[v])$. The q be an integer where the above procedure stops, which is the case when G_q has no edges. Notice that for any $i \in [q]_0$, each $S \in \{S_j \backslash S_i \mid j \in \{i+1, i+2, \cdots, q\}\}$ is a pair of perfectly matched sets in G_i. The above in particular implies that S_q is a pair of perfectly matched sets in $G = G_0$. Also, for each $i \in [q]_0$, $|S_i| = i$. If $q \geqslant k$, then we have obtained a pair of perfectly matched sets in G of size at least k, and thus we can conclude that the instance is a yes-instance. Otherwise $q \leqslant k-1$, and we let $Q = \{u_i, v_i \mid i \in [q]\}$. Consider any vertex $u \in V(G) \backslash N_G[Q]$. Since G has no isolated vertices, u must have a neighbor v in G. Note that $v \notin Q$, as $u \in V(G) \backslash N_G[Q]$. Also, if $v \notin N_G(Q)$, then $\{u,v\}$ is an edge in G_q, which contradicts that G_q has no edges. The above discussions imply that Q is a 2-dominating set in G of size at most $|Q| \leqslant 2 \cdot (k-1)$. ◀

We are now ready to prove Theorem 2.

Proof of Theorem 2. Consider an instance (G,k) of $\mathcal{H}$-Minor Free PMS. If Lemma 12 returns that the instance is a yes-instance, then we are done. Otherwise, it returns a 2-dominating set in G of size at most $2 \cdot (k-1)$. From Proposition 10, the treewidth of G is bounded by $c_{\mathcal{H}} \cdot \sqrt{2 \cdot (k-1)}$, where $c_{\mathcal{H}}$ is a constant depending on the family $\mathcal{H}$. Now using Lemma 7.4 of [8], we compute a nice tree decomposition of width at most $c_{\mathcal{H}} \cdot \sqrt{2 \cdot (k-1)}$ in time bounded by $\mathcal{O}(nk)$. Now we can use Proposition 11 to resolve the instance. ◀

5 FPT Algorithm for $K_{b,b}$-free Graphs

The goal of this section is to prove Theorem 3. Consider any fixed number $b \in \mathbb{N}$. Recall that a graph is $K_{b,b}$-free if it does not contain a *subgraph* isomorphic to $K_{b,b}$. We obtain an FPT algorithm for Perfectly Matched Sets on $K_{b,b}$-free graphs by using an approach similar to random separation [3], in combination with the below-stated result of Dabrowski et al. [9].

▶ **Proposition 13** (Lemma 2, [9]). *For any natural numbers s, t and p, there is a number $N'(s,t,p)$ such that every graph with a matching of size at least $N'(s,t,p)$ contains either a clique K_s, an induced bi-clique $K_{t,t}$ or an induced matching of size p. Here, $N'(s,t,p) = R(s, R(s, N(t,p)))$ where $R(s,t)$ is the non-symmetric Ramsey number.*

Let (G,k) be an instance of Perfectly Matched Sets, where G is a $K_{b,b}$-free graph with n vertices. We color the vertices of $V(G)$ independently and randomly using two colors, *red* and *blue* (with equal probability). This forms a random partition $V_R \uplus V_B$ of the vertices of G, where V_R and V_B are the set of vertices colored with red and blue color, respectively. We call these two partitions as *color classes*. Next, we obtain the graph G' from G by removing all the edges between the vertices of the same color class. Thus, the edges in G' have endpoints of differing colors, and thus it is bipartite. We compute (in polynomial time) a maximum sized matching M in G' [23]. We will next argue that either M has at most $N'(3,b,k)$ edges, or we can conclude that (G,k) is a yes-instance.

Case 1. Firstly suppose that M has at least $N'(3,b,k)$ edges. Recall that G is bipartite, so it does not have any K_3. Moreover, as G is $K_{b,b}$-free, we can obtain that G' has no induced $K_{b,b}$. As the size of a maximum matching in G' is at least $N'(3,b,k)$, using Proposition 13 we can obtain that G' has an induced matching M_I of size at least k. Now using the next observation we can conclude that (G,k) is a yes-instance of the problem.

▶ **Observation 14.** *$(V_R \cap V(M_I), V_B \cap V(M_I))$ is a pair of perfectly matched sets in G of size at least k.*

Proof. Consider $x \in V_R \cap V(M_I)$, where x has a neighbor $y \in V_B \cap V(M_I)$ and $\{x,y\}$ is an edge in M_I. Let $z \neq y$ be another neighbor of x in $V_B \cap V(M_I)$. Then since x is colored with red and z is colored with blue, the edge $(x,z) \in E(G')$. But this is a contradiction to the fact that M_I is an induced matching. From the above discussions, we can obtain that each vertex in $V_R \cap V(M_I)$ has exactly one neighbor in $V_B \cap V(M_I)$ and vice-versa. ◀

Case 2. Now suppose that in G' the matching M has less than $N'(3,b,k)$ edges, and thus, $\mathrm{tw}(G') \leqslant 2 \cdot N'(3,b,k)$. Now in G', we look for a pair of perfectly matched sets (X,Y) where $X \subseteq V_R$ and $Y \subseteq V_B$. Let us denote this version of Perfectly Matched Sets as the *colored*-Perfectly Matched Sets problem. Aravind et al. [1] designed an FPT algorithm for Perfectly Matched Sets parameterized by the treewidth of the given graph. They use a *nice* tree decomposition of the graph, where in each bag $\beta(t)$, $X \cap \beta(t)$ and $Y \cap \beta(t)$ play a crucial role in the construction of their algorithm. To adapt their algorithm for *colored*-Perfectly Matched Sets, we only need to enforce that $X \cap \beta(t)$ and $Y \cap \beta(t)$ are selected from V_R and V_B, respectively. Precisely in Section 5.3 of their draft [1], $A \cap \beta(t) = A_t$

and $B \cap \beta(t) = B_t$ can be replaced by $A \cap (\beta(t) \cap V_R) = A_t$ and $B \cap (\beta(t) \cap V_B) = B_t)$, respectively, to obtain an algorithm for the colored version. Notice that they denote the desired perfectly matched sets by (A, B) while we do it by (X, Y). Hence we have an FPT algorithm running in time $2^{\mathcal{O}(\text{tw}(G'))} \cdot n^{\mathcal{O}(1)}$ to obtain a pair of perfectly matched sets (X, Y) of G' of size k where $X \subseteq V_R$, $Y \subseteq V_B$. We remark that the algorithm given by [1] can actually compute such a set by the standard backtracking technique, and thus even for our colored case, we can compute a pair of perfectly matched sets in G'. Now we claim the following.

▶ **Observation 15.** *(X, Y) is also a pair of perfectly matched sets of G.*

Proof. Suppose (X, Y) is not a pair of perfectly matched sets of G. Notice that $E(G') \subseteq E(G)$ and hence there is a vertex v in X with more than one neighbor in Y or there is a vertex u in Y with more than one neighbor in X. Without loss of generality let such a vertex v be in X. Let two of its neighbors in Y be y_1 and y_2. But the edges $\{v, y_1\}$ and $\{v, y_2\}$ are also in G' as they have endpoints with differing colors. But this contradicts the fact that (X, Y) is a pair of perfectly matched sets of G'. ◀

In the construction of G' from G, we delete edges with endpoints in the same color classes. Hence a pair of perfectly matched sets of G may not remain a pair of perfectly matched sets of G'. But in the claim below, we show that for a fixed size of perfectly matched sets, the chances of such an event happening stays low.

▶ **Observation 16.** *Any k-sized perfectly matched sets (X, Y) of G is also a perfectly matched sets of G' with probability at least 2^{-2k}.*

Proof. The probability that all vertices of X are colored red and all vertices of Y are colored blue is at least 2^{-2k}. Thus we can obtain that with probability at least 2^{-2k} (X, Y) is also a perfectly matched sets of G'. ◀

The proof of the following lemma follows from Observations 14, 15 and 16 with the standard trick of making independent runs of the discussed algorithm.

▶ **Lemma 17.** *There exists a randomized FPT algorithm running in time $2^{\mathcal{O}(N'(3,b,k)+k)} \cdot n^{\mathcal{O}(1)}$ that, given a* Perfectly Matched Sets *instance (G, k) on $K_{b,b}$-free graphs, either reports a failure or finds a pair of perfectly matched sets in G of size at least k. Moreover, if the algorithm is given a yes-instance, it returns a solution with constant probability.*

We now explain the derandomization procedure for the above algorithm. It involves deterministically constructing a family $\mathcal{F}$ of coloring functions $f : [n] \to [2]$ rather than selecting a random coloring $\chi : [n] \to [2]$ such that it is assured that one of the functions from $\mathcal{F}$ colors one set from a pair of perfectly matched sets of size k (when (G, k) is a yes-instance) with color 1 and the other set with color 2. To this end, we will use the following.

▶ **Definition 18** (Definition 5.19, [8])**.** An (n, k)-*universal set* is a family $\mathcal{U}$ of subsets of $[n]$ such that for each $S \subseteq [n]$ of size k, the family $\{A \cap S : A \in \mathcal{U}\}$ contains all 2^k subsets of S.

▶ **Proposition 19** (Theorem 5.20, [8])**.** *For any $n, k \geqslant 1$, we can construct an (n, k)-universal set of size $2^k k^{\mathcal{O}(\log k)} \log n$ in time $2^k k^{\mathcal{O}(\log k)} n \log n$.*

We assume that $V(G) = [n]$ (otherwise we can relabel the vertices). We first construct an $(n, 2k)$-universal set, $\mathcal{U}$, using the above proposition. Now we construct a family of function $\mathcal{F}$ from $[n]$ to $\{1, 2\}$ as follows, where $\mathcal{F}$ is initialized to $\varnothing$. For each $U \in \mathcal{U}$, add the function

$f_U : [n] \rightarrow [2]$, where $f^{-1}(1) = U$. Note that if G has a pair of perfectly matched sets (A, B) of size k, then there is $U \in \mathcal{U}$, such that $(A \cup B) \cap U = A$. Thus at least one function in $\mathcal{F}$ is the correct coloring for us. We can iterate over each of the colorings given by $\mathcal{F}$, and this leads us to the following result.

▶ **Theorem 20.** *Perfectly Matched Sets on $K_{b,b}$-free graphs admits a deterministic FPT algorithm running in time $k^{\mathcal{O}(\log k)} \cdot 2^{\mathcal{O}(N'(3,b,k)+k)} \cdot n^{\mathcal{O}(1)}$.*

6 Kernelization for Perfectly Matched Sets on d-degenerate graphs

In this section, we design a polynomial kernel for d-degenerate graphs, and thus prove Theorem 4. We design our kernel using the *strong systems of distinct representatives* [17] (to be defined shortly). Recall that a graph G is d-degenerate if every induced subgraph of it contains a vertex of degree at most d. We start by stating the definition of strong systems of distinct representatives and a useful result regarding it.

▶ **Definition 21** (Strong systems of distinct representatives, [18])**.** A k-tuple $(x_1, x_2, \ldots, x_k)$ is a *system of distinct representatives* for sets $S_1, S_2, \ldots, S_k$, if for each $i \in [k]$, $x_i \in S_i$. Moreover, it is *strong* if additionally, for each $i \in [k]$ and $j \in [k] \setminus \{i\}$, $x_i \notin S_j$.

▶ **Proposition 22** (Theorem 8.12 [17])**.** *Consider any family $\mathcal{F}$ with more than $\binom{r+k}{k}$ distinct sets of sizes at most r. Then, at least $k+2$ sets in this family have a strong system of distinct representatives.*

The following property of a d-degenerate graph follows directly from the definition.

▶ **Proposition 23.** *A d-degenerate graph on n vertices has at most dn edges.*

Next, we give a lower bound on the number of low-degree vertices in a d-degenerate graph.

▶ **Lemma 24.** *Let G be d-degenerate graph with $n \geqslant 6$ vertices. Then G has strictly more than $5n/6$ vertices of degree at most $12d$.*

Proof. Let G be d-degenerate graph with n vertices. By Proposition 23, the number of edges in G is at most dn. So the sum of the degrees of the vertices in G is bounded by $2dn$. Assume that, there are at most $5n/6$ vertices of degree at most $12d$ in G. Then we have a set $U \subseteq V(G)$ of at least $n/6 \geqslant 1$ vertices of degree strictly more than $12d$. Now the sum of the degrees of the vertices in U is strictly more than $(n/6) \cdot 12d = 2dn$, a contradiction. Hence there are strictly more than $5n/6$ vertices of degree at most $12d$ in G. ◀

▶ **Observation 25.** *In a pair of perfectly matched sets (A, B) of a graph G, there are at most two non-adjacent vertices $x, y \in A \cup B$ such that $N(x) = N(y)$.*

Proof. Let $x, y, z \in A \cup B$ be three pairwise non-adjacent vertices such that $N(x) = N(y) = N(z)$. At least two of these vertices are either in A or B. Without loss of generality let $x, y \in A$. But then x and y, both have the exactly same neighbors in B, which contradicts that $A \cup B$ is a pair of perfectly matched sets of G. ◀

With Observation 25, we obtain the following reduction rule.

▶ **Reduction Rule 1.** *Let u, v, w be three distinct vertices in $V(G)$ such that $N(u) = N(v) = N(w)$, then reduce (G, k) to $(G - w, k)$.*

▶ **Lemma 26.** *Reduction Rule 1 is safe.*

Proof. Consider an application of Reduction Rule 1 in which a vertex, say $w \in V(G)$ was deleted because there are two distinct vertices u and v other than w such that $N(u) = N(v) = N(w)$. We will prove that (G, k) is a yes-instance of PERFECTLY MATCHED SETS if and only if $(G - w, k)$ is a yes-instance of PERFECTLY MATCHED SETS.

If $(G - w, k)$ is a yes-instance, any pair of perfectly matched sets in $G - w$ is also a pair of perfectly matched sets in G, thus (G, k) must also be a yes-instance. For the other direction suppose that (G, k) is a yes-instance of the problem, and we have two disjoint sets $A, B \subseteq V(G)$ such that every vertex in A has exactly one neighbor in B and vice-versa. If $w \notin A \cup B$, then (A, B) is a pair of perfectly matched sets in $G - w$ of size k, and we are done. Else, exactly one of A and B must contain w. Without loss of generality we assume that $w \in A$. From Observation 25, we know that $|(A \cup B) \cap \{u, v, w\}| \leqslant 2$. Now neither v nor u belongs to A. If $B \cap \{u, v\} = \varnothing$, then $(A \backslash \{w\} \cup \{u\}, B)$ is a pair of perfectly matched sets in $G - w$ of size k. Else, exactly one of v or u belongs to B, say $u \in B$ (the other case is symmetric). Then, $(A \backslash \{w\} \cup \{v\}, B)$ is a pair of perfectly matched sets in $G - w$ of size k. ◀

We are now ready to prove Theorem 4.

Proof of Theorem 4. Let (G, k) be an instance of PERFECTLY MATCHED SETS where G is a d-degenerate graph. If Reduction Rule 1 on (G, k) is applicable, then we apply it in polynomial time and reduced the number of vertices. When the reduction rule is no longer applicable, we do the following. Let X be the set of vertices of with degree at most $12d$, and let $t = |X|$. Consider the family $\mathcal{F} = \{N(u) \mid u \in X\}$ (with repetitions removed). By the non-applicability of Reduction Rule 1 and Lemma 24, we can obtain that $|\mathcal{F}| \geqslant t/2 \geqslant (5n/6)/2 = 5n/12$. Also note that each set in $\mathcal{F}$ has size at most $12d$.

If $|\mathcal{F}| \leqslant \binom{12d+k}{k}$, then $5n/12 < \mathcal{F} \leqslant \binom{12d+k}{k}$. Therefore n, i.e., the number of vertices in G is bounded by $k^{O(d)}$. Otherwise, $|\mathcal{F}| > \binom{12d+k}{k}$, and we argue that (G, k) is a yes-instance. From Proposition 22, at least $k + 2$ of these sets form $\mathcal{F}$ have a strong system of distinct representatives, say these sets are $N(v_1), N(v_2), \cdots, N(v_{k+2})$ and $(u_1, u_2, \cdots, u_{k+2})$ is its strong system of distinct representatives. Let $A = \{v_1, v_2, \cdots, v_{k+2}\}$ and $B = \{u_1, u_2, \cdots, v_{u+2}\}$. Note that for each $i \in [k + 2]$, we have $\{v_i, u_i\} \in E(G)$. For any $i \in [k + 2]$ and $j \in [k] \backslash \{i\}$, $\{v_i, u_j\} \notin E(G)$, as $u_j \notin N(v_i)$ by the definition of a strong system of distinct representatives. Thus, (A, B) is a pair of perfectly matched sets of size at least $(k + 2)$ in G. ◀

As planar graphs are 5-degenerate, the above result directly gives us a polynomial kernel (which is not linear!) for planar graphs. We next obtain a linear kernel for planar graphs.

Linear Kernel on Planar Graphs. We describe a procedure to obtain a linear-sized vertex kernel for planar graphs. To this end, we state the following useful result.

▶ **Proposition 27** (Theorem 4.11, [18]). *A twinless planar graph with $n \geqslant 2$ vertices contains an induced matching of size at least $n/40$.*

From Proposition 27, we have the following observation.

▶ **Observation 28.** *Let G be a planar graph on $n \geqslant 4$ vertices such that there are no three vertices that are pairwise false twins. Then G contains a pair of perfectly matched sets of size at least $n/80$.*

Proof. From G, we can construct a twinless planar graph G' by keeping exactly one of the false twins i.e. for any two false twins u and v, we delete exactly one of them. Hence G' is a twinless planar graph with size at least $n/2 \geqslant 2$ vertices. From Proposition 27, G' has an induced matching of size at least $n/80$, which is also an induced matching in G. But such an induced matching gives us a pair of perfectly matched sets of size $n/80$. ◀

▶ **Theorem 29.** Perfectly Matched Sets *on planar graphs admits an* $\mathcal{O}(k)$*-sized kernel.*

Proof. Consider an instance (G, k) of the problem, where G is a planar graph with n vertices. Apply Reduction Rule 1 as long as it is applicable. If $|V(G)| < 2$, then we are done. Otherwise, from Observation 28, G has a pair of perfectly matched sets with size at least $n/80$. If $k \leqslant n/80$, then the given instance is a yes-instance, and otherwise $|V(G)| < 80k$. ◀

References

1 N. R. Aravind and Roopam Saxena. Perfectly Matched Sets in Graphs: Hardness, Kernelization Lower Bound, and FPT and Exact Algorithms. *CoRR*, abs/2107.08584, 2021. arXiv:2107.08584.

2 Paul S. Bonsma. The complexity of the matching-cut problem for planar graphs and other graph classes. *J. Graph Theory*, 62(2):109–126, 2009.

3 Leizhen Cai, Siu Man Chan, and Siu On Chan. Random Separation: A New Method for Solving Fixed-Cardinality Optimization Problems. In Hans L. Bodlaender and Michael A. Langston, editors, *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, volume 4169 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2006.

4 Kathie Cameron. Induced matchings in intersection graphs. *Discret. Math.*, 278(1-3):1–9, 2004.

5 Yixin Cao and Dániel Marx. Interval Deletion Is Fixed-Parameter Tractable. *ACM Trans. Algorithms*, 11(3):21:1–21:35, 2015.

6 Chi-Yeh Chen, Sun-Yuan Hsieh, Hoàng-Oanh Le, Van Bang Le, and Sheng-Lung Peng. Matching Cut in Graphs with Large Minimum Degree. *Algorithmica*, 83(5):1238–1255, 2021.

7 Vasek Chvátal. Recognizing decomposable graphs. *J. Graph Theory*, 8(1):51–53, 1984.

8 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

9 Konrad K. Dabrowski, Marc Demange, and Vadim V. Lozin. New results on maximum induced matchings in bipartite graphs and beyond. *Theor. Comput. Sci.*, 478:33–40, 2013.

10 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Dimitrios M. Thilikos. The Bidimensional Theory of Bounded-Genus Graphs. *SIAM J. Discret. Math.*, 20(2):357–371, 2006.

11 Reinhard Diestel. Graph theory. *Graduate texts in mathematics*, 173, 2017.

12 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

13 Shimon Even, Oded Goldreich, Shlomo Moran, and Po Tong. On the NP-completeness of certain network testing problems. *Networks*, 14(1):1–24, 1984.

14 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

15 Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Subexponential algorithms for partial cover problems. *Inf. Process. Lett.*, 111(16):814–818, 2011.

16 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *CoRR*, abs/1311.3899, 2013. arXiv:1311.3899.

17 Stasys Jukna. *Extremal Combinatorics – With Applications in Computer Science*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2011.

18 Iyad A. Kanj, Michael J. Pelsmajer, Marcus Schaefer, and Ge Xia. On the induced matching problem. *J. Comput. Syst. Sci.*, 77(6):1058–1070, 2011.

19 Johannes Köbler, Sebastian Kuhnert, and Osamu Watanabe. Interval graph representation with given interval and intersection lengths. *Journal of Discrete Algorithms*, 34:108–117, 2015.

20 Łukasz Kowalik, Matjaž Krnc, Tomasz Waleń, et al. Improved induced matchings in sparse graphs. *Discrete Applied Mathematics*, 158(18):1994–2003, 2010.

21 Hoàng-Oanh Le and Van Bang Le. On the Complexity of Matching Cut in Graphs of Fixed Diameter. In *27th International Symposium on Algorithms and Computation, ISAAC 2016, December 12-14, 2016, Sydney, Australia*, volume 64 of *LIPIcs*, pages 50:1–50:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.

22 Van Bang Le and Jan Arne Telle. The Perfect Matching Cut Problem Revisited. In *Graph-Theoretic Concepts in Computer Science – 47th International Workshop, WG 2021, Warsaw, Poland, June 23-25, 2021, Revised Selected Papers*, volume 12911 of *Lecture Notes in Computer Science*, pages 182–194. Springer, 2021.

23 Silvio Micali and Vijay V. Vazirani. An O(sqrt(|v|) |E|) Algorithm for Finding Maximum Matching in General Graphs. In *21st Annual Symposium on Foundations of Computer Science, Syracuse, New York, USA, 13-15 October 1980*, pages 17–27. IEEE Computer Society, 1980.

24 Hannes Moser and Somnath Sikdar. The parameterized complexity of the induced matching problem. *Discret. Appl. Math.*, 157(4):715–727, 2009.

25 Augustine M. Moshi. Matching cutsets in graphs. *J. Graph Theory*, 13(5):527–536, 1989.

26 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.

27 Larry J. Stockmeyer and Vijay V. Vazirani. NP-Completeness of Some Generalizations of the Maximum Matching Problem. *Inf. Process. Lett.*, 15(1):14–19, 1982.

28 Michele Zito. Induced matchings in regular graphs and trees. In *Graph-Theoretic Concepts in Computer Science*, pages 89–101, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.