The Parameterized Complexity of the Induced Matching Problem [2,3]

Hannes Moser [1,*]

Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany

Somnath Sikdar

The Institute of Mathematical Sciences, C.I.T Campus, Taramani, Chennai 600113, India

Abstract

Given a graph G and an integer $k \geq 0$, the NP-complete Induced Matching problem asks whether there exists an edge subset M of size at least k such that M is a matching and no two edges of M are joined by an edge of G. The complexity of this problem on general graphs as well as on many restricted graph classes has been studied intensively. However, little is known about the parameterized complexity of this problem.

As the problem is W[1]-hard in general graphs, we study the parameterized complexity for several restricted graph classes. In this work, we provide first-time fixed-parameter tractability results for planar graphs, bounded-degree graphs, graphs with girth at least six, bipartite graphs, line graphs, and graphs of bounded treewidth. In particular, we give a linear-size problem kernel for planar graphs.

Key words: Induced Matching, Parameterized Complexity, Planar Graph, Kernelization, Tree Decomposition

* Corresponding author. Fax: +49 3641 9-46002, Phone: +49 3641 9-46324
Email addresses: `moser@minet.uni-jena.de` (Hannes Moser), `somnath@imsc.res.in` (Somnath Sikdar).

1 Supported by the Deutsche Forschungsgemeinschaft, project ITKO (iterative compression for solving hard network problems), NI 369/5.

2 Supported by a DAAD-DST exchange program, D/05/57666.

3 An extended abstract of this work appears under the title "The Parameterized Complexity of the Induced Matching Problem in Planar Graphs" in the proceedings of the 2007 International Frontiers of Algorithmics Workshop (FAW'07), Springer, LNCS 4613, pages 325–336, held in Lanzhou, China, August 01–03, 2007.

1 Introduction

A *matching* in a graph is a set of edges no two of which have a common endpoint. An *induced matching* M of a graph $G = (V, E)$ is an edge-subset $M \subseteq E$ such that M is a matching and no two edges of M are joined by an edge of G. In other words, the set of edges of the subgraph induced by $V(M)$ is precisely the set M. The size of a maximum induced matching in G is denoted by $\mathrm{im}(G)$. The decision version of Induced Matching is defined as follows.

Input: An undirected graph $G = (V, E)$ and a nonnegative integer k.
Question: Does G have an induced matching with at least k edges?

The optimization version asks for an induced matching of maximum size.

The Induced Matching problem was introduced as a variant of the maximum matching problem and motivated by Stockmeyer and Vazirani [42] as the "risk-free" marriage problem [4] . This problem has been intensively studied in recent years. It is known to be NP-complete for the following graph classes (among others):

- planar graphs of maximum degree 4 [32],
- bipartite graphs of maximum degree 3, C_4-free bipartite graphs [34],
- r-regular graphs for $r \geq 5$, line-graphs, chair-free graphs, and Hamiltonian graphs [33].

The problem is known to be polynomial time solvable for the following graph classes:

- trees [22,43],
- chordal graphs [8],
- weakly chordal graphs [10],
- circular arc graphs [23],
- trapezoid graphs, interval-dimension graphs, and comparability graphs [24],
- interval-filament graphs, polygon-circle graphs, and AT-free graphs [9],
- (P_5,D_m)-free graphs [33,35],
- (P_k,$K_{1,n}$)-free graphs [35],
- (bull, chair)-free graphs, line-graphs of Hamiltonian graphs [33],
- and graphs where the maximum matching and the maximum induced matching have the same size [33].

For graphs in which the maximum matching and maximum induced matching have the same size, Cameron and Walker [11] provide a simple characterization

[4] Find the maximum number of pairs such that each married person is compatible with no married person except the one he or she is married to.

Graph Class	Param.	Result	Reference
general	k	$W[1]$-hard	[36]
bounded degree	k	$O(k)$ kernel	Sect. 3.1
bipartite	k	$W[1]$-hard	Sect. 3.3
graphs with girth at least 6	k	$O(k^3)$ kernel	Sect. 3.2
line graphs	k	$O^*(2^k)$ alg.	Sect. 3.4
planar	k	$O(k)$ kernel	Sect. 4
bounded treewidth	ω	$O(4^\omega \cdot n)$ alg.	Sect. 5

Fig. 1. Parameterized complexity results for NP-complete variants of Induced Matching. Here, k denotes the minimum number of edges of the induced matching, and ω denotes the treewidth of the input graph.

of these graphs which also leads to a simpler recognition algorithm.

Regarding polynomial-time approximability, it is known that Induced Matching is APX-complete on r-regular graphs, for all $r \geq 3$, and bipartite graphs with maximum degree 3 [17]. Moreover, for r-regular graphs it is NP-hard to approximate Induced Matching to within a factor of $r/2^{O(\sqrt{\ln r})}$ [13]. In general graphs, the problem cannot be approximated to within a factor of $n^{1/2-\epsilon}$ for any $\epsilon > 0$, where n is the number of vertices of the input graph [38]. There exists an approximation algorithm for the problem on r-regular graphs ($r \geq 3$) with asymptotic performance ratio $r-1$ [17], which has subsequently been improved to $0.75r + 0.15$ [25]. Moreover, there exists a polynomial-time approximation scheme (PTAS) for planar graphs of maximum degree 3 [17].

In contrast to these results, little is known about the parameterized complexity of Induced Matching. To the best of our knowledge, the only known result is that the problem is $W[1]$-hard (with respect to the matching size as parameter) in the general case [36], and hence unlikely to be fixed-parameter tractable. Therefore, it is of interest to study the parameterized complexity of the problem in those restricted graph classes where it remains NP-complete.

An interesting aspect of studying the parameterized complexity of NP-complete problems are problem kernels. The intuitive idea behind kernelization is that a polynomial-time preprocessing step removes the “easy” parts of a problem instance such that only the “hard” core of the problem remains, which can then be solved by other methods. We call such a core a linear kernel if we can prove that its size is a linear function of the parameter k. Linear problem kernels are of immense interest in parameterized algorithmics. One can consult the recent surveys by Fellows [19], Guo and Niedermeier [26], and the books by Flum and Grohe [20] and Niedermeier [37] for an overview on kernelization.

In this paper we give linear kernels for planar graphs and bounded-degree graphs. For graphs of girth at least 6, which also include C_4-free bipartite graphs, we can show a simple kernel with a cubic number of vertices (that is, $O(k^3)$ vertices). Moreover, we show that Induced Matching is fixed-parameter tractable for line graphs. Finally, we give an algorithm for graphs of bounded treewidth using an improved dynamic programming approach, which runs in $O(4^\omega \cdot n)$ time, where ω is the width of the given tree decomposition. This extends an algorithm for Induced Matching on trees by Zito [43]. See Figure 1 for an overview of the results presented in this paper.

Our main result, the linear kernel on planar graphs, is based on a kernelization technique first introduced by Alber et al. [3] to show that Dominating Set has a linear kernel on planar graphs. The result for the kernel size has subsequently been improved by Chen et al. [12], and they also show lower bounds on the kernel size for Dominating Set, Vertex Cover, and Independent Set on planar graphs. The technique developed by Alber et al. [3] has been exploited by Guo et al. [28] in developing a linear kernel for Full-Degree Spanning Tree, a maximization problem. Moreover, Fomin and Thilikos [21] extended the technique to graphs of bounded genus. Recently, Guo and Niedermeier [27] gave a generic kernelization framework for NP-hard problems on planar graphs based on that technique. So far, the technique has been applied to problems whose solutions are vertex subsets. Our linear kernel on planar graphs is the first application of this technique for a maximization problem whose solutions are edge subsets. We adapt and extend the technique introduced in [3] and [28]. Note that very recently our kernelization result on planar graphs has been improved by Kanj et al. to a kernel of $40k$ vertices using a different technique [29].

The paper is organized as follows. First we define our notation in Section 2. In Section 3 we give the results for bounded-degree graphs, graphs of girth at least 6, bipartite graphs, and line graphs. These results are simple and meant to provide some first-time insight into the parameterized complexity of Induced Matching on these classes. We then give a linear problem kernel on planar graphs in Section 4, which is the most technical part of this paper. Finally, we give the improved dynamic programming algorithm for graphs of bounded treewidth in Section 5.

2 Preliminaries

In this paper, we deal with fixed-parameter algorithms that emerge from the field of parameterized complexity analysis [16,20,37], where the computational complexity of a problem is analyzed in a two-dimensional framework. One dimension of an instance of a parameterized problem is the input size n, and the other is the *parameter* k. A parameterized problem is *fixed-parameter tractable* if it can be solved in $f(k) \cdot n^{O(1)}$ time, where f is a computable function depending only on the parameter k.

A common method to prove that a problem is fixed-parameter tractable is to provide *data reduction rules* that lead to a *problem kernel*. Given a problem instance (I, k), a data reduction rule replaces that instance by an equivalent instance (I', k') in polynomial time such that $|I'| \leq |I|$ and $k' \leq k$. Two problem instances are *equivalent* if they are both YES-instances or both NO-instances. An instance to which none of a given set of data reduction rules applies is called *reduced* with respect to that set of rules. A parameterized problem is said to have a problem kernel if, after the application of the data reduction rules, the resulting reduced instance has size $f(k)$ for a function f depending only on k. A kernel is called *linear* if its size is linear in k, that is, if $f(k) = c \cdot k$ for some constant c. Analogous to classical complexity theory, Downey and Fellows [16] developed a framework providing a reducibility and completeness program. The basic complexity class for fixed-parameter intractability is $W[1]$ as there is good reason to believe that $W[1]$-hard problems are not fixed-parameter tractable [16].

In this paper we assume that all graphs are simple and undirected. For a graph $G = (V, E)$, we write $V(G)$ to denote its vertex set and $E(G)$ to denote its edge set. By default, for a given graph we use n and m to denote the number of vertices and edges, respectively. A vertex that is an endpoint of an edge is *incident* to that edge and *adjacent* to the other endpoint. An *isolated* vertex has no neighbors. For a subset $V' \subseteq V$, by $G[V']$ we mean the subgraph of G induced by V'. We write $G \setminus V'$ to denote the graph $G[V \setminus V']$. For a vertex $v \in V$ we also write $G - v$ instead of $G \setminus \{v\}$. The *open neighborhood* $N(W)$ of a vertex set W is the set of all vertices in $V \setminus W$ that are adjacent to some vertex in W. The *closed neighborhood* $N[W]$ is defined as $N(W) \cup W$. For a vertex v we write $N(v)$ ($N[v]$) instead of $N(\{v\})$ ($N[\{v\}]$).

We assume that paths are *simple*, that is, a vertex is contained at most once in a path. A path P from a to b is denoted as a vector $P = (a, \ldots, b)$, and a and b are called the *endpoints* of P. The *length* of a path $(a_1, a_2, \ldots, a_q)$ is $q-1$, that is, the number of edges on it. For an edge set M we define $V(M) := \bigcup_{e \in M} e$. The *distance* $d(u, v)$ between two vertices u, v is the length of a shortest path between them. The *distance* between two edges e_1, e_2 is the minimum distance between two vertices $v_1 \in e_1$ and $v_2 \in e_2$.

If a graph can be drawn on the plane without edge crossings then it is *planar*. A *plane* graph is a planar graph with a fixed embedding in the plane. Given a plane graph, a cycle $C = (a, \ldots, a)$ of length at least three encloses an *area* A of the plane. The cycle C is called the *boundary* of A, all vertices in the area A are *inside* A. A vertex is *strictly inside* A if it is inside A and not on C.

3 Fundamental Results

The following results are basic first-time fixed-parameter tractability results for several graph classes where Induced Matching remains NP-hard.

3.1 Bounded-Degree Graphs

We show that Induced Matching admits a linear problem kernel on graphs whose maximum degree is bounded by a constant.

Proposition 1 *The* Induced Matching *problem admits a problem kernel of $O(k \cdot d^2)$ vertices on graphs whose vertex degrees are bounded by d (that is, the kernel is linear for constant d). The kernel can be obtained in $O(n)$ time.*

PROOF. Let G be a graph with maximum degree d, where d is some constant. Let M be any maximal induced matching of G found by the following greedy algorithm. The algorithm repeatedly selects an arbitrary edge e, adds it to the solution, and deletes $N[V(e)]$. This process is repeated until no more edges remain. Since the maximum degree of the graph is bounded by d, selecting an edge and deleting its closed neighborhood takes constant time only, and the process is repeated at most $\lfloor n/2 \rfloor$ times, thus the whole greedy algorithm runs in $O(n)$ time.

If $|M| \geq k$, then we are done. Therefore, assume that $|M| < k$. Define S_1 and S_2 as follows: $S_1 := N(V(M))$ and $S_2 := N(S_1) \setminus V(M)$. Note that all neighbors of vertices in S_2 are in the set S_1, since if a vertex $u \in S_2$ has a neighbor $v \notin S_1$ then $\{u, v\}$ could be added to the induced matching, contradicting its maximality. Clearly, $|S_1| < 2kd$ and $|S_2| < 2kd^2$. Since $V(G) = V(M) \cup S_1 \cup S_2$, it immediately follows that $|V(G)| < 2k(1 + d + d^2)$. □

3.2 Graphs Without Small Cycles

As stated before, the Induced Matching problem is NP-hard on C_4-free bipartite graphs [34]. Since the class of C_4-free bipartite graphs is properly contained in the class of graphs with girth at least six, the Induced Matching problem is NP-hard on the latter graph class.

Proposition 2 *The* Induced Matching *problem admits a problem kernel of $O(k^3)$ vertices on graphs with girth at least six. The corresponding data reduction rule can be carried out in $O(n + m)$ time.*

PROOF. Let G be a graph with girth at least 6. Delete all except one degree-one neighbor from as many vertices as possible in G. If every vertex has degree at most k then we obtain a kernel of $O(k^3)$ vertices immediately from Proposition 1. Therefore assume that there exists a vertex u of degree at least $k+1$. Let $S := \{v_1, \ldots, v_{k+1}\}$ be a set of $k+1$ neighbors of u. Since G has no 3-cycles, S is independent. At most one vertex of S has degree one. Assume without loss of generality that the vertices in $\{v_1, \ldots, v_k\}$ have degree at least two. For $1 \leq i < j \leq k$, v_i and v_j do not have any common neighbors as otherwise we obtain a 4-cycle. For $1 \leq i \leq k$, let z_i be a neighbor of v_i. Again $\{z_1, \ldots, z_k\}$ must be independent as otherwise we obtain a 5-cycle. But then $\{(v_1, z_1), \ldots, (v_k, z_k)\}$ is an induced matching of size k. $\square$

The fact that many $W[1]$-hard problems become fixed-parameter tractable in graphs with no small cycles was discovered by Raman and Saurabh [40].

3.3 Bipartite Graphs

For bipartite graphs we show that the Induced Matching problem is $W[1]$-hard. We give a reduction from the $W[1]$-complete Irredundant Set problem [15]. Given a graph $G = (V, E)$ and a positive integer k, Irredundant Set asks whether there exists a set $V' \subseteq V$ of size at least k having the property that each vertex $u \in V'$ has a private neighbor. A *private neighbor* of a vertex $u \in V'$ is a vertex $u' \in N[u]$ (possibly $u' = u$) such that for every vertex $v \in V' \setminus \{u\}, u' \notin N[v]$.

Proposition 3 *The* Induced Matching *problem in bipartite graphs is* $W[1]$-*hard with respect to the parameter* k.

PROOF. We prove the proposition by a reduction from Irredundant Set. Let (G, k) be an instance of the Irredundant Set problem. Construct a bipartite graph G' as follows. Construct two copies of the vertex set of G and call these V' and V''; the copies of a vertex $u \in V(G)$ from V' and V'' are denoted as u' and u'', respectively. Define $V(G') = V' \cup V''$ and $E(G') = \{\{u', u''\} : u \in V(G)\} \cup \{\{u', v''\}, \{v', u''\} : \{u, v\} \in E(G)\}$. We claim that the graph G has an irredundant set of size k if and only if G' has an induced matching of size k. To show the claim, suppose $S = \{w_1, \ldots, w_k\} \subseteq V(G)$ is an irredundant set of size k in G. For $1 \leq i \leq k$, let x_i be the private neighbor of w_i. Then for all i, $\{w_i', x_i''\}$ is an edge in G'. Since the x_i's are private neighbors there is no edge $\{w_i, x_j\}$ in G for all $i \neq j$ and therefore no edge $\{w_i', x_j''\}$ in G'. Therefore, the edges $\{w_i', x_i''\}, \ldots, \{w_k', x_k''\}$ form an induced matching in G'. Conversely, if $M = \{e_1, \ldots, e_k\}$ is an induced matching of G' of size k then for each $e_i = \{u_i', v_i''\}$, $v_i'' \in V''$ can be viewed as a private neighbor of $u_i' \in V'$.

Therefore, the vertices $u_1, \ldots, u_k$ form an irredundant set in G. This completes the proof. $\square$

3.4 Line Graphs

The line graph $L(G)$ of a graph G is defined as follows: the vertex set of $L(G)$ is the edge set of G; two "vertices" e_1 and e_2 of $L(G)$ are connected by an edge if e_1 and e_2 share an endpoint. More formally, we have

$$L(G) := (E(G), \{\{e_1, e_2\} : e_1, e_2 \in E(G) \wedge e_1 \cap e_2 \neq \emptyset\}).$$

A graph H is a line graph if there exists a graph G such that $H = L(G)$. It is well-known (see, e.g., [18]) that if H is a line graph, then it does not have any induced $K_{1,3}$ (also known as *claw*). It was shown that the INDUCED MATCHING problem is NP-complete on line graphs (and hence claw-free graphs) [33].

Given a graph H, it is possible to test in time $\max\{|V(H)|, |E(H)|\}$ whether H is a line-graph and if so construct G such that $H = L(G)$ [41].

Lemma 4 *Let H be a line-graph and let $H = L(G)$. Then H has an induced matching of size at least k if and only if G has at least k vertex-disjoint copies (not necessarily induced) of P_3, the path on three vertices.*

PROOF. Let $\{e_1, \ldots, e_k\}$ be an induced matching of size k in H. From the definition of a line-graph it follows that each edge e_i corresponds to a path $p_i = (x_i, y_i, z_i)$ in the graph G. The set $\cup_{i=1}^{k}\{x_i, y_i, z_i\}$ has exactly $3k$ vertices. Moreover, the sets $\{x_i, y_i, z_i\}$ and $\{x_j, y_j, z_j\}$ are disjoint for $i \neq j$: if any two vertices, one from path p_i and the other from p_j, are identical, then an endpoint of e_i would be connected to an endpoint of e_j, contradicting the fact that these edges form an induced matching. This shows that G contains k vertex-disjoint copies of P_3. Conversely, if G has k vertex-disjoint copies of P_3, then the edges corresponding to these paths form an induced matching in H. $\square$

The problem of checking whether a given graph G has k copies of P_3 can be solved in $O(2^{5.301k}k^{2.5}+n^3)$ time and is therefore fixed-parameter tractable [39]. (A more general method to solve such kind of packing problems can be found in [31].)

Proposition 5 *The* INDUCED MATCHING *problem on line-graphs can be solved in time $O(2^{5.301k}k^{2.5} + n^3)$ and is therefore fixed-parameter tractable.*

4 A Linear Kernel on Planar Graphs

In order to show our kernel, we employ the following data reduction rules. These rules stem from the simple observation that if two vertices have the same neighborhood, one of them can be removed without affecting the size of a maximum induced matching. Compared to the data reduction rules applied in other proofs of planar kernels [3,12,28], these data reduction rules are quite simple and can be carried out in $O(n+m)$ time on general graphs (and hence in $O(n)$ time on planar graphs).

(R0) *Degree Zero Rule:* Delete vertices of degree zero.
(R1) *Degree One Rule:* If a vertex u has two distinct neighbors x, y of degree 1, then delete x.
(R2) *Degree Two Rule:* If u and v are two vertices such that $|N(u) \cap N(v)| \geq 2$ and if there exist two vertices $x, y \in N(u) \cap N(v)$ with $\deg(x) = \deg(y) = 2$, then delete x.

Note that these data reduction rules are parameter-independent. The following lemma is easy to show.

Lemma 6 *The data reduction rules $R0, R1$, and $R2$ are correct.*

PROOF. Obviously none of these rules destroys planarity. The correctness of the Degree Zero Rule is obvious since no isolated vertex can be part of an edge. Concerning the Degree One Rule, observe that only one edge incident to u can be part of an induced matching. The correctness of the Degree Two Rule can be seen as follows. Let G be a graph and M a maximum induced matching for G. If one of the vertices x or y is an endpoint of an edge in M, then either u or v is the other endpoint of that edge since x and y have no other neighbors. Suppose, without loss of generality, that $\{u, x\}$ is a matching edge. Since u and y are adjacent, y cannot be an endpoint of an edge in M, and since x is adjacent to v, v cannot be an endpoint of an edge in M. For that reason, we can get a new matching $M' := (M \setminus \{u, x\}) \cup \{\{u, y\}\}$, which has the same size as M and is still induced, and it is an induced matching for $G' := G - x$. The case where no vertex in $\{x, y\}$ is an endpoint of an edge in M is obvious. The reverse direction is trivial, as any induced matching M' for G' is also an induced matching for G. □

Lemma 7 *The data reduction rules $R0, R1$, and $R2$ can be carried out in $O(n)$ time on planar graphs and $O(n+m)$ time on general graphs, where n and m denote, respectively, the number of vertices and edges.*

PROOF. We first remove all isolated vertices in $O(n)$ time in order to reduce the graph with respect to the Degree Zero Rule. Then we apply the Degree Two Rule. For each vertex u of the graph we check which neighbors of u can be deleted. To this end, we determine in $O(\deg(u))$ time all degree-two neighbors of u; then we group together all such neighbors whose second neighbor is the same. For each group, we mark all but one vertex for deletion. After having done this for every vertex we delete the marked vertices. Finally we apply the Degree One Rule. For each vertex u we determine in $O(\deg(u))$ time all degree-one neighbors of u, and delete all but one. The running time to exhaustively apply each rule is $O(\sum_{u \in V}(1 + \deg(u)))$, which is bounded by $O(n + m)$ for general graphs and $O(n)$ for planar graphs.

It remains to explain why we need to check every vertex for each rule only once, and why we first apply the Degree Two Rule and then the Degree One Rule. It is easy to verify that for each rule the following holds: a vertex that is not deleted during the application of the rule does not become a candidate for deletion with respect to the rule *after* the application of that rule. Moreover, we have to justify why we apply the Degree Two Rule before the Degree One Rule. If the Degree Two Rule cannot be applied anymore, then the application of the Degree One Rule cannot cause any situation where the Degree Two Rule could be applied again. This does not hold if we apply the rules the other way around. The application of the Degree Zero Rule at the beginning is obviously correct. □

The following theorem is our main result whose proof spans the remainder of this section.

Theorem 8 *Let $G = (V, E)$ be a planar graph reduced with respect to the rules $R0$, $R1$, and $R2$. Then $|V| \leq c \cdot \mathrm{im}(G)$ for some constant c. That is, the* Maximum Induced Matching *problem on planar graphs admits a linear problem kernel.*

The basic observation is that if M is a maximum induced matching of a graph $G = (V, E)$ then for each vertex $v \in V$ there exists a $u \in V(M)$ such that $d(u, v) \leq 2$. Otherwise, we could add edges to M and obtain a larger induced matching. Since every vertex in the graph is within distance at most two to some vertex in $V(M)$, we know, roughly speaking, that the edges in M have distance at most four to other edges in M. This leads to the idea of regions "in between" matching edges that are close to each other. We will see that these regions cannot be too large if the graph is reduced with respect to the above data reduction rules. Moreover, we show that there cannot be many vertices that are not contained within such regions.

This idea of a region decomposition was introduced in [3], but the definition of a region as it appears there is much simpler since the regions are defined between vertices, and they are smaller. The remaining part of this section is dedicated to the proof of Theorem 8. First, in Section 4.1 we show how to find a "maximal region decomposition" of a reduced graph that contains only $O(|M|)$ regions, where M is the size of a maximum induced matching of the graph. Then, in Section 4.2 we show that a region in such a maximal region decomposition contains only a constant number of vertices. Finally, in Section 4.3 we show that in any reduced graph there are only $O(|M|)$ vertices which lie outside of regions.

4.1 Finding a Maximal Region Decomposition

Definition 9 *Let G be a plane graph and M a maximum induced matching of G. For edges $e_1, e_2 \in M$, a* region *$R(e_1, e_2)$ is a closed subset of the plane such that*

(1) the boundary of $R(e_1, e_2)$ is formed by two length-at-most-four paths
- *$(a_1, \ldots, a_2)$, $a_1 \neq a_2$, between $a_1 \in e_1$ and $a_2 \in e_2$,*
- *$(b_1, \ldots, b_2)$, $b_1 \neq b_2$, between $b_1 \in e_1$ and $b_2 \in e_2$, and*

by e_1 if $a_1 \neq b_1$ and e_2 if $a_2 \neq b_2$;

(2) for each vertex x in the region $R(e_1, e_2)$, there exists a $y \in V(\{e_1, e_2\})$ such that $d(x, y) \leq 2$;

(3) no vertices inside the region other than endpoints of e_1 and e_2 are from M.

The set of boundary vertices of R is denoted by δR. We write $V(R(e_1, e_2))$ to denote the set of vertices of a region $R(e_1, e_2)$, that is, all vertices strictly inside $R(e_1, e_2)$ together with the boundary vertices δR. A vertex in $V(R(e_1, e_2))$ is inside *R.*

Note that the two enclosing paths may be identical; the corresponding region then consists solely of a simple path of length at most four. Note also that e_1 and e_2 may be identical. For an example of a region see Figure 2.

Definition 10 *Let G be a plane graph and M a maximum induced matching in G. An M-*region decomposition *of $G = (V, E)$ is a set $\mathcal{R}$ of regions such that no vertex in V lies strictly inside more than one region from $\mathcal{R}$. For an M-region decomposition $\mathcal{R}$, we define $V(\mathcal{R}) := \bigcup_{R \in \mathcal{R}} V(R)$. An M-region decomposition $\mathcal{R}$ is* maximal *if there is no $R \notin \mathcal{R}$ such that $\mathcal{R} \cup \{R\}$ is an M-region decomposition with $V(\mathcal{R}) \subsetneq V(\mathcal{R}) \cup V(R)$.*

For an example of an M-region decomposition, see Fig. 3.

Lemma 11 *Given a plane reduced graph $G = (V, E)$ and a maximum induced*

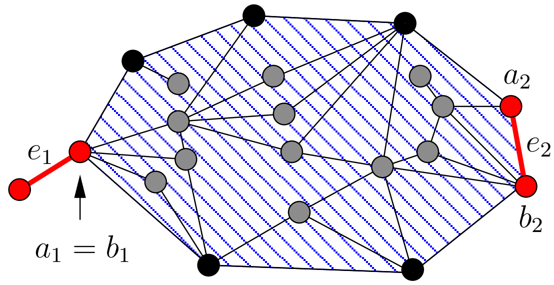

Fig. 2. Example of region $R(e_1, e_2)$ between two edges $e_1, e_2 \in M$. Note that e_1 is not part of R, but only its endpoint $a_1 = b_1$. The black vertices are the boundary vertices, and the gray vertices in the hatched area are the vertices strictly inside of R.

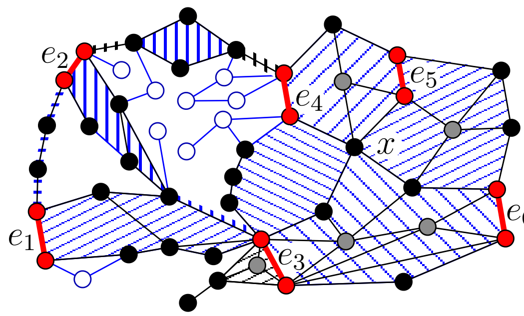

Fig. 3. An example of an M-region decomposition: black vertices denote boundary vertices; gray vertices lie strictly inside a region and white vertices lie outside of regions. Each region is hatched with a different pattern. Note the special cases, as for instance regions that consist of a path like the region between e_1 and e_2, or regions that are created by only one matching edge (the region on the left side of e_3). Note also that boundary vertices may be contained in boundaries of several regions, that is, the boundaries may touch each other. See for instance vertex x as an example of a boundary vertex of four regions.

matching M of G, there exists an algorithm that constructs a maximal M-region decomposition with $O(|M|)$ regions.

PROOF. We use a constructive proof with a greedy algorithm as shown in Figure 4. This algorithm is quite similar to the algorithms by Alber et al. [3] and Guo et al. [28] used for their linear kernel for Dominating Set on planar graphs and Full-Degree Spanning Tree on planar graphs, respectively. A similar algorithm is also applied in [27].

It is clear that the algorithm returns an M-region decomposition. To see that the returned M-region decomposition $\mathcal{R}$ is maximal, observe that for every vertex u that is not in a region we check whether there is a region containing u that can be added to $\mathcal{R}$. It remains to show that $\mathcal{R}$ contains $O(|M|)$ regions. The proof of this is similar to the proof by Alber et al. [3] and is not given in full detail here.

Algorithm: Maximum M-region decomposition.
Input: A graph $G = (V, E)$ and a maximum induced matching M.
Output: An M-region decomposition $\mathcal{R}$ with $O(|M|)$ regions.

$\mathcal{R} \leftarrow \emptyset, V' \leftarrow \emptyset$
for each vertex $u \in V$ **do**
    if $u \notin V'$ and there exists a region $R(e_1, e_2)$ with $u \in V(R(e_1, e_2))$
      such that $\mathcal{R} \cup \{R\}$ is an M-region decomposition **then**
        $S \leftarrow$ set of all regions $R(e_1, e_2)$ with $u \in V(R(e_1, e_2))$
          such that $\mathcal{R} \cup \{R\}$ is an M-region decomposition
        $R_{new} \leftarrow$ region from S that is space-maximal
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{R_{new}\}, V' \leftarrow V' \cup V(R_{new})$
    end if
end for
return $\mathcal{R}$

Fig. 4. A greedy algorithm that constructs an M-region decomposition for a plane graph G and a maximum induced matching M.

The main idea is to show that between any two edges e_1 and e_2 of a maximum induced matching M there is a constant number of regions. To show that the number of regions is $O(|M|)$, construct a new graph by replacing the edges of the induced matching by vertices and the regions by edges; that is, place an edge between two vertices in the new graph if there exists a region between the corresponding edges in the original graph. The resulting graph is a planar multigraph and by Euler's formula there are at most $c \cdot (3|M| - 6)$ edges, where c is the maximum number of regions between two edges e_1, e_2 of the original graph. This proves that the number of regions in the original graph is indeed $O(|M|)$. □

4.2 *Bounding the Size of a Region*

To upper-bound the size of a region R we make use of the fact that any vertex strictly inside R has distance at most two from some vertex in δR. For this reason, the vertices strictly inside R can be arranged in two layers. The first layer consists of the neighbors of boundary vertices, and the second of all the remaining vertices, that is, all vertices at distance at least two from every boundary vertex. The proof strategy is to show that if any of these layers contains too many vertices, then there exists an induced matching M' with $|M'| > |M|$. An important structure for our proof are areas enclosed by 4-cycles, called *diamonds*.

Definition 12 *Let u and v be two vertices in a plane graph. A* diamond[5] *is a*

[5] In standard graph theory, a diamond denotes a 4-cycle with exactly one chord.

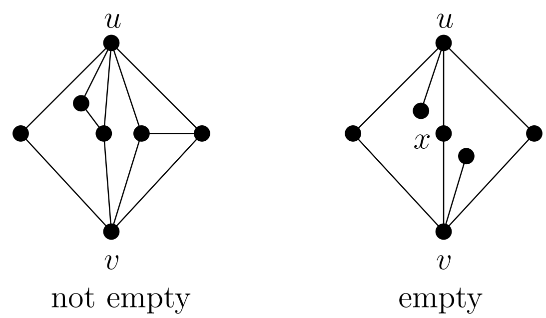

Fig. 5. A diamond (left) and an empty diamond (right) in a reduced plane graph.

closed area of the plane with two length-2 paths between u and v as boundary. A diamond $D(u, v)$ is empty*, if every edge e in the diamond is incident to either u or v.*

Fig. 5 shows an empty and a non-empty diamond. In a reduced plane graph empty diamonds have a restricted size. We are especially interested in the maximum number of vertices strictly inside an empty diamond $D(u, v)$ that have both u and v as neighbors.

Lemma 13 *Let $D(u, v)$ be an empty diamond in a reduced plane graph. Then there exists at most one vertex strictly inside $D(u, v)$ that has both u and v as neighbors.*

PROOF. Suppose that there are at least two vertices x and y strictly inside $D(u, v)$, where both have u and v as neighbors. Since D is empty, x and y can have no other neighbors than u and v. Thus, there are two vertices of degree two with the same neighbors, a contradiction to the fact that G is reduced (Degree Two Rule). □

Lemma 13 shows that if there are more than three edge-disjoint length-two paths between two vertices u, v, then there must be an edge e in an area enclosed by two of these paths such that e is not incident to u or v. This fact is used in the following lemma to show that the number of length-two paths between two vertices of a reduced plane graph is bounded.

Lemma 14 *Let u and v be two vertices of a reduced plane graph G such that there exists two distinct length-2 paths (u, x, v) and (u, y, v) enclosing an area A of the plane. Let M be a maximum induced matching of G. If neither x nor y is an endpoint of an edge in M and no vertex strictly inside A is contained in $V(M)$, then the following holds:*

We abuse this term here. Note that diamonds also play an important role in proving linear problem kernels on planar graphs for other problems [3,27].

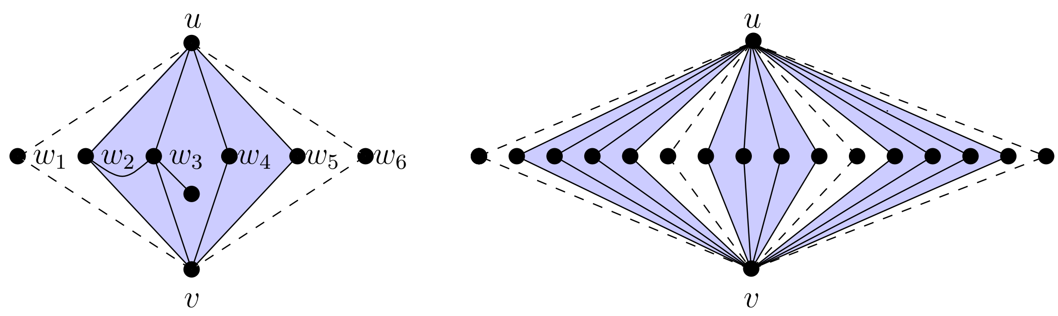

Fig. 6. Left: An embedding of the vertices $w_1, \ldots, w_6$ for the first case in the proof of Lemma 14. Right: An embedding of 16 neighbors of u and v for the last case of the proof. The diamonds are shaded and the "isolation paths" are drawn with dashed lines.

If neither u nor v is an endpoint of an edge in M, then there are at most 5 *edge-disjoint length-2 paths between u and v inside A. If exactly one of u or v is an endpoint of an edge in M, then there are at most* 10 *such paths, and if both u and v are endpoints of edges in M, then there are at most* 15 *such paths.*

PROOF. The idea is to show that if there are more than the claimed number of length-2 paths between u and v, then we can exhibit an induced matching M' with $|M'| > |M|$, which would then contradict the optimality of M.

First, we consider the case when neither u nor v is contained in $V(M)$. Suppose for the purpose of contradiction that there are 6 common neighbors $w_1, \ldots, w_6$ of u and v that lie inside A (that is, strictly inside and on the enclosing paths). Without loss of generality, suppose that these vertices are embedded as shown in Fig. 6 (left-hand side), with w_1 and w_6 lying on the enclosing paths. Consider the diamond D with the boundary induced by the vertices u, v, w_2, w_5. Since w_3 and w_4 are strictly inside D and are incident to both u and v, by Lemma 13, we know that D is not empty. That is, there exists an edge e in D which is not incident to u or v. Clearly e is incident to neither w_1 nor w_6 and the endpoints of e are at distance at least 2 from every vertex in $V(M)$. Therefore, we can add e to M and obtain a larger induced matching, which contradicts the optimality of M.

Next, consider the case when exactly one of u or v is an endpoint of an edge e in M. Using the same idea as above, it is easy to see that if there exist 11 length-2 paths between u and v, then there are at least two non-empty diamonds (using (u, w_1, v), (u, w_6, v) and (u, w_{11}, v) as "isolation paths") whose boundaries share only u and v. We can then replace e in M by edges e_1 and e_2, one from each nonempty diamond, and obtain a larger induced matching.

The last case, when both u and v are endpoints of edges in M, can be handled in the same way by showing that there exist at least three non-empty diamonds if we assume 16 length-2 paths between u and v, where the boundaries of these diamonds only touch in u and v (see Figure 6). Then we can replace the edges in M that are incident to u and v by three edges strictly inside the diamonds, contradicting the maximum cardinality of M. □

Lemma 14 is needed to upper-bound the number of vertices inside and outside of regions that are connected to at least two boundary vertices.

The next two lemmas are needed to upper-bound the number of vertices that are connected to exactly one boundary vertex. First, Lemma 15 upper-bounds the number of such vertices under the condition that they are contained in an area which is enclosed by a short cycle. Lemma 15 is then used in Lemma 16 to upper-bound the total number of such vertices for a given boundary vertex.

Lemma 15 *Let u be a vertex in a reduced plane graph G and let $v, w \in N(u)$ be two distinct vertices that have distance at most three in $G - u$. Let P denote a shortest path between v and w in $G - u$ and let A denote the area of the plane enclosed by P and the path (v, u, w). If there are at least 9 neighbors of u strictly inside A, then there is at least one edge strictly inside A.*

PROOF. Let u contain nine neighbors $\{z_1, \ldots, z_9\}$ strictly inside A and assume that there is no edge strictly inside A. By the Degree One Rule, at most one of the z_i's can have degree 1. Without loss of generality assume that z_9 has degree 1. By the Degree Two Rule, no two degree-2 vertices have the same neighborhood. Observe that the neighbors of the z_i's must be vertices on P due to planarity, as otherwise there would be an edge strictly inside of A, a contradiction to our assumption.

First, consider the case when there exists a vertex among the z_i's of degree at least 4. Suppose z_j, $1 \leq j \leq 8$, has at least three neighbors among the vertices in P. Because the graph is planar, there exists a $x \in P$ such that no z_i, $i \neq j$, is adjacent to x. The remaining vertices have degree 2 or 3 and each is adjacent to some vertex $y \neq x$ in P. Moreover, there can be at most one vertex of degree 3. Since $|V(P)| \leq 4$, it is easy to see that there are at least two degree-2 vertices with the same neighbors, a contradiction.

Therefore, assume that $\deg(z_i) \leq 3$ for all i. Again by planarity, there are at most three vertices in $\{z_1, \ldots, z_8\}$ of degree 3. The remaining at least five vertices must be of degree 2 and each is adjacent to a vertex in P. Since $|V(P)| \leq 4$, this implies that there are two degree-2 vertices with the same neighborhood, a contradiction. This shows that if there exist nine neighbors of u in A, there exists an edge strictly inside A. □

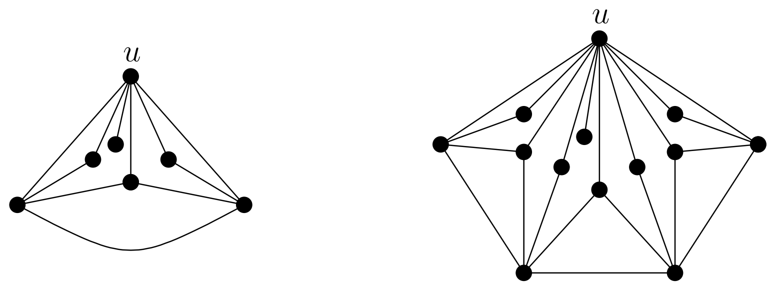

Fig. 7. Worst-case embeddings to illustrate Lemma 15.

Fig. 7 shows, for two different situations, the maximum number of neighbors of u that can be strictly inside A such that no edge lies strictly inside A.

Lemma 16 *Let G be a reduced plane graph, let M be a maximum induced matching of G, let $e_1, e_2 \in M$ be edges that form a region $R(e_1, e_2)$, and let u be a boundary vertex of R. Then, u has at most 40 neighbors strictly inside R that are not adjacent to any other boundary vertex.*

PROOF. We assume that there are 41 neighbors of u strictly inside R that are not adjacent to any other boundary vertex and show that then we can find an induced matching M' with $|M'| > |M|$, contradicting the maximum cardinality of M.

Suppose that the neighbors $v_1, \ldots, v_{41}$ are embedded around u in a clockwise fashion. By the Degree One Rule, u can have at most one neighbor of degree 1. Without loss of generality assume that $\deg(v_2) = 1$. Consider the vertices v_1, v_{11}, and v_{21}. If the pairwise distance of these vertices in $G-u$ is at least four, then any three edges e_a, e_b, e_c in $G-u$ incident to v_1, v_{11}, and v_{21}, respectively, are pairwise non-adjacent. Since they lie strictly inside $R(e_1, e_2)$ (u is the only neighbor on the boundary), we can set $M' := (M \setminus \{e_1, e_2\}) \cup \{e_a, e_b, e_c\}$. Similarly if v_{21}, v_{31}, and v_{41} have a pairwise distance of at least four, then we can construct an induced matching of cardinality larger than $|M|$.

It remains to show the case that at least two vertices from $\{v_1, v_{11}, v_{21}\}$ have distance at most three and at least two vertices from $\{v_{21}, v_{31}, v_{41}\}$ have distance at most three. Let $\{w_1, w_1'\} \subseteq \{v_1, v_{11}, v_{21}\}$ and $\{w_2, w_2'\} \subseteq \{v_{21}, v_{31}, v_{41}\}$ be these vertices. Let P_1 and P_2 denote, respectively, the shortest paths from w_1 to w_1' and from w_2 to w_2' in $G-u$. Note that P_1 and P_2 are strictly inside R. Let A_1 be the area enclosed by P_1 and the path (w_1, u, w_1') and let A_2 be the area enclosed by P_2 and the path (w_2, u, w_2'). Note that P_1 and P_2 can be chosen so that the subsets of the plane strictly inside A_1 and A_2 do not intersect. By Lemma 15, there exists edges e_1, e_2 such that e_1 is strictly inside A_1 and e_2 is strictly inside A_2. If there exists an edge $e \in M$ incident to u, then $(M - e) \cup \{e_1, e_2\}$ is an induced matching with size strictly larger than that of M, a contradiction. If no edge of M is incident to u, $M \cup \{e_1, e_2\}$

is again an induced matching of larger size. □

Using Lemma 14 and Lemma 16, we can now upper-bound the number of vertices inside a region.

Lemma 17 *Each region $R(e_1, e_2)$ of an M-region decomposition of a reduced plane graph contains $O(1)$ vertices.*

PROOF. We prove the lemma by partitioning the vertices strictly inside $R(e_1, e_2)$ into two sets A and B, where A consists of all vertices at distance exactly one from some boundary vertex, and B consists of all vertices at distance at least two from every boundary vertex, and then showing that $|A|$ and $|B|$ are upper-bounded by a constant.

To this end, partition A into A_1 and A_2, where A_1 contains all vertices in A that have exactly one neighbor on the boundary, and A_2 all vertices that have at least two neighbors on the boundary. To upper-bound the size of A_1, observe that due to Lemma 16, a vertex $v \in \delta R$ on the boundary can have at most 40 neighbors in A_1. Since a region has at most ten boundary vertices, we conclude that A_1 contains at most 400 vertices.

Next we upper-bound the size of A_2. Consider the planar graph G' induced by $\delta R \cup A_2$. Every vertex in A_2 is adjacent to at least two boundary vertices in G'. Replace every vertex $v \in A_2$ with an edge connecting two arbitrary neighbors of v on the boundary. Merge multiple edges between two boundary vertices into a single edge. Since G' is planar, the resulting graph must also be planar. As $|\delta R| \leq 10$, using the Euler formula we conclude that the resulting graph has at most $3 \cdot 10 - 6 = 24$ newly added edges. By Lemma 14, each such edge represents at most 15 length-two paths, and thus $|A_2| \leq 24 \cdot 15 = 360$.

To upper-bound the size of B, observe that $G[B]$ must be a graph without edges (that is, B is an independent set). By the Degree One Rule, each vertex in A has at most one neighbor in B of degree one. Therefore, there are $O(1)$ degree-one vertices in B. To bound the number of degree-at-least-two vertices in B, we use the same argument as the one used to bound the size of A_2. Since $|A| = O(1)$, there is a constant number of degree-at-least-two vertices in B. Therefore $|B| = O(1)$. This completes the proof. □

Proposition 18 *Let G be a reduced plane graph and let M be a maximum induced matching of G. There exists an M-region decomposition such that the total number of vertices inside all regions is $O(|M|)$.*

PROOF. Using Lemma 11, there exists a maximal M-region decomposition for G with at most $O(|M|)$ regions. By Lemma 17, each region has a constant number of vertices. Thus there are $O(M)$ vertices inside regions. □

We next bound the number of vertices that lie outside regions of a maximal M-region decomposition.

4.3 Bounding the Number of Vertices Lying Outside of Regions

In this section, we upper-bound the number of vertices that lie outside of regions of a maximal M-region decomposition. The strategy to prove this bound is similar to that used in the last section. We subdivide the vertices lying outside of regions into several disjoint subsets and upper-bound their sizes separately.

Note again that the distance from any vertex of the graph to a vertex in $V(M)$ is at most two. We partition the vertices lying outside of regions into two sets A and B, where A is the set of vertices at distance exactly one from some vertex in $V(M)$, and B is the set of vertices at distance at least two from every vertex in $V(M)$. We bound the sizes of these two sets separately.

Partition A into two subsets A_1 and A_2, where A_1 is the set of vertices that have exactly one boundary vertex as neighbor, and A_2 is the set of vertices that have at least two boundary vertices as neighbors. Note that each vertex v in A can be adjacent to exactly one vertex $u \in V(M)$. For if it is adjacent to distinct vertices $u, w \in V(M)$, then the path (u, v, w) can be added to the region decomposition, contradicting its maximality (recall that regions can consist of simple paths between two vertices in $V(M)$). To bound the number of vertices in A_1 we need the following lemma.

Lemma 19 *Let v be a vertex in A_1 and let u be its neighbor in $V(M)$. Then for all $w \in V(M) \setminus \{u\}$, the distance between v and w in $G - u$ is at least three.*

PROOF. Let u and v be as in the statement of the Lemma and let $w \in V(M) \setminus \{u\}$. Suppose (v, x, w) is a path of length two. Now x cannot be a boundary vertex since $v \in A_1$. The path $P = (u, v, x, w)$ is of length three and the only vertices of P that are boundary vertices are u and w. Thus P can be added in the region decomposition, contradicting its maximality. □

Lemma 20 *Given a maximal M-region decomposition consisting of $O(|M|)$ regions, the set A contains $O(|M|)$ vertices.*

PROOF. To bound the size of A_1, we claim that each vertex $u \in V(M)$ has at most 20 neighbors in A_1. Suppose, for the purpose of contradiction, that 21 vertices $v_1, \ldots, v_{21}$ in A_1 are adjacent to $u \in V(M)$. Also assume that they are embedded in a clockwise fashion around u in that order. Let e be the edge in M incident to u. First, suppose that v_1 and v_{11} have distance at least four in $G - u$. Then there exist edges e_a, e_b in $G - u$ incident to v_1 and v_{11}, respectively, that form an induced matching of size 2. Moreover by Lemma 19, the endpoints of e_a and e_b are not adjacent to any vertex of $V(M)$ in $G - u$. Therefore, $M' = (M \setminus \{e\}) \cup \{e_a, e_b\}$ is an induced matching of size larger than that of M, a contradiction to the maximum cardinality of M. The same holds if the distance between v_{11} and v_{21} is at least four in $G - u$. Therefore assume that in the graph $G - u$, $d(v_1, v_{11}) \leq 3$ and $d(v_{11}, v_{21}) \leq 3$. Let P_1 and P_2 be shortest paths in $G - u$ between v_1 and v_{11} and between v_{11} and v_{21}, respectively. Note that due to Lemma 19 these two paths cannot contain any vertex from $V(M)$. By Lemma 15, the areas enclosed by P_1 and (v_1, u, v_{11}), and P_2 and v_{11}, u, v_{21}, respectively, contain an edge strictly inside them. The edge e can be replaced by these two edges to obtain an induced matching of size larger than M, a contradiction to the maximum cardinality of M. This proves our claim. Since there are exactly $2\,|M|$ vertices in $V(M)$, this shows that the total number of vertices in A_1 is at most $40\,|M|$.

Next, we bound the size of A_2. Every vertex v in A_2 is adjacent to a vertex $u \in V(M)$ and some boundary vertex $w \notin V(M)$. Vertex w must be adjacent to u, for otherwise there is a path consisting of the vertices (u, v, w) and some subpath on the boundary where w lies which can be added to the region decomposition $\mathcal{R}$, contradicting its maximality. Since there are $O(|M|)$ regions, there are $O(|M|)$ possible boundary vertices adjacent to a vertex in $V(M)$. By Lemma 14, given a vertex $x \in V(M)$ and $y \in V \setminus V(\mathcal{R})$ there can be at most 10 vertices adjacent to both x and y. This shows that A_2 contains $O(|M|)$ vertices. □

It remains to bound the number of vertices in B, that is, the number of vertices outside of regions that are at distance at least two from every vertex in $V(M)$.

Lemma 21 *Given a maximal M-region decomposition with $O(|M|)$ regions, the set B contains $O(|M|)$ vertices.*

PROOF. To bound the size of B, observe that $G[B]$ is a graph without edges. Furthermore, observe that $N(B) \subseteq A \cup A'$, where A' is the set of boundary vertices in the M-region decomposition that are different from $V(M)$. By Lemma 20 and since the boundary of each region contains a constant number of vertices, the set $C := A \cup A'$ contains $O(|M|)$ vertices.

First, consider the vertices in B that have degree one. Obviously, there can be at most $|C|$ such vertices due to the Degree One Rule. The remaining vertices are adjacent to at least two vertices in C. We can use an argument similar to the one used in the proof of Lemma 17 (using the Euler formula) to show that there are $O(|C|)$ degree-at-least-two vertices in B. Thus, $|B| = O(|C|) = O(|M|)$. □

Using these results, we can see that the total number of vertices outside of regions is bounded.

Proposition 22 *Given a maximal M-region decomposition with $O(|M|)$ regions, the number of vertices that lie outside of regions is $O(|M|)$.*

PROOF. The proof directly follows from Lemmas 20 and 21. □

Using Propositions 18 and 22, we can show that, given a reduced plane graph G and a maximum induced matching M of G, there exists an M-region decomposition with $O(|M|)$ regions such that the number of vertices inside and outside of regions is $O(|M|)$. This shows the $O(|M|)$ upper bound on the number of vertices as claimed in Theorem 8, that is, Maximum Induced Matching admits a linear problem kernel on planar graphs.

5 Induced Matching on Graphs with Bounded Treewidth

Zito [43] developed a linear-time dynamic programming algorithm to solve Induced Matching on trees. We extend his work and obtain a linear-time algorithm on graphs of bounded treewidth [7]. Note that compared to Zito's work our dynamic programming approach uses a different encoding to store the partial solutions in the updating process.

It is relatively easy to verify that such a linear-time algorithm for graphs of bounded treewidth actually does exist.

Proposition 23 *Let $\omega \geq 1$. Given a graph with a tree decomposition of width at most ω, the* Maximum Induced Matching *problem can be decided in linear time.*

PROOF. We give a monadic-second order logic (MSO) formulation of Max-

IMUM INDUCED MATCHING:

$$\max E' : \forall e_1 \forall e_2 \Big(E'e_1 E'e_2 \neg \Big[\exists x \exists y V x \wedge V y \wedge I x e_1 \wedge I y e_2 \wedge ((x = y) \vee \exists e' (E e' \wedge I x e' \wedge I y e')) \Big] \Big)$$

In the above formula, V and E are unary relation symbols which denote the vertex and edge set of the graph; I is a binary relation symbol that denotes whether a vertex is incident to an edge and E' denotes an induced matching. One can now use Courcelle's result [14] which states that all graph properties definable in monadic second-order logic can be decided in linear time on graphs of bounded treewidth. □

Courcelle's result is purely theoretical as the hidden constants in the running time analysis are huge. As such, it is of independent interest to develop algorithms which can be used in practice.

It is relatively easy to see that a standard dynamic programming approach would result in a running time of $O(9^\omega \cdot n)$, where ω is the width of the given tree-decomposition. With an improved dynamic programming algorithm, we obtain a running time of $O(4^\omega \cdot n)$. Our approach also uses some ideas that were applied for an improved dynamic programming algorithm for DOMINATING SET [1,4]. However, the concept of monotonicity which was needed for DOMINATING SET is not needed for INDUCED MATCHING, as the necessary condition for an improved analysis of the dynamic programming update process is fulfilled without the monotonicity concept. Here we describe only the basic definitions and those parts of the algorithm which are important in showing the improved running time. We also refer the reader to the standard literature about tree decompositions [5–7,30].

Definition 24 *Let $G = (V, E)$ be a graph. A* tree decomposition *of G is a pair $(\{X_i \mid i \in I\}, T)$, where each X_i is a subset of V, called a* bag, *and T is a tree with the elements of I as* nodes. *The following three properties must hold:*

(1) $\bigcup_{i \in I} X_i = V$,
(2) for every edge $e \in E$ there is an $i \in I$ such that $e \subseteq X_i$, and
(3) for all $i, j, k \in I$, if j lies on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.

The width *of $(\{X_i \mid i \in I\}, T)$ equals* $\max\{|X_i| \mid i \in I\} - 1$. *The* treewidth *of G is the minimum k such that G has a tree decomposition of width k.*

A tree decomposition with a simpler structure is defined as follows.

Definition 25 *A tree decomposition $(\{X_i \mid i \in I\}, T)$ is called a* nice tree decomposition *if the following conditions are satisfied (we suppose the decomposition tree T to be rooted at some arbitrary but fixed node):*

(1) Every node of the tree T has at most two children.
(2) If a node i has two children j and k, then $X_i = X_j = X_k$ (in this case i is called a join node*).*
(3) If a node i has one child j, then either
   (a) $|X_i| = |X_j| + 1$ and $X_j \subset X_i$ (in this case i is called an introduce node*), or*
   (b) $|X_i| = |X_j| - 1$ and $X_i \subset X_j$ (in this case i is called a forget node*).*

A given tree decomposition can be transformed into a nice tree decomposition in linear time:

Lemma 26 (Lemma 13.1.3 of [30]) *Given a tree decomposition of a graph G that has width ω and $O(n)$ nodes, where n is the number of vertices of G. Then we can find a nice tree decomposition of G that also has width ω and $O(n)$ nodes in time $O(n)$.*

The remainder of this section is dedicated to the proof of the following theorem.

Theorem 27 *Let $G = (V, E)$ be a graph with a given nice tree decomposition $(\{X_i \mid i \in I\}, T)$. Then the size of a maximum induced matching of G can be computed in $O(4^\omega \cdot n)$ time, where $n := |I|$ and ω denotes the width of the tree decomposition.*

PROOF. For each bag X_i we consider all possible ways of obtaining an induced matching in the subgraph induced by X_i and all bags below X_i. To do this, we create a table $A_i, i \in I$ for each bag X_i which stores this information. These tables are updated in a bottom-up process starting at the leaves of the decomposition tree. In the following, we say that a vertex v is *contained* in an induced matching M if v is an endpoint of an edge in M. If v is contained in M, its *partner* in M is a vertex u such that $\{u, v\} \in M$. We use different colors to represent the possible states of a vertex in a bag:

white(0): A vertex labeled 0 is not contained in M.
black(1): A vertex labeled 1 is contained in M and its partner in M has already been discovered in the current stage of the algorithm.
gray(2): A vertex labeled 2 is contained in M but its partner in M has not been discovered in the current stage of the algorithm.

For each bag $X_i = \{x_{i_1}, \ldots, x_{i_{n_i}}\}$, $|X_i| = n_i$, we construct a table A_i consisting of 3^{n_i} rows and $n_i + 1$ columns. Each row represents a coloring $c : X_i \to \{0, 1, 2\}^m$ of the graph $G[X_i]$; the entry $m_i(c)$ in the n_i+1st column represents the number of vertices in an induced matching in the graph visited up to the current stage of the algorithm under the assumption that the vertices in the bag X_i are assigned colors as specified by c. If no induced matching is possible with the corresponding coloring, then the entry $m_i(c)$ stores the value $-\infty$. For a coloring $c = (c_1, \ldots, c_m) \in \{0, 1, 2\}^m$ and a color $d \in \{0, 1, 2\}$ we define $\#_d(c) := |\{1 \leq t \leq m \mid c_t = d\}|$.

Given a bag X_i and a coloring c of the vertices in X_i, we say that c is *valid* if the subgraph induced by the vertices labeled 1 and 2 has the following structure: vertices labeled 2 have degree 0 and those labeled 1 have either degree 0 or 1. For valid colorings we store the value m_i as described above; for all other colorings we set m_i to $-\infty$ to mark it as invalid. A coloring is *strictly valid* if it is valid and, in addition, vertices labeled 1 induce isolated edges. We next describe the dynamic programming process. Recall that we assume that we work with a nice tree decomposition.

Leaf Nodes

For a leaf node X_i compute the table A_i as

$$m_i(c) := \begin{cases} \#_1(c) + \#_2(c), & \text{if } c \text{ is strictly valid,} \\ -\infty, & \text{otherwise.} \end{cases}$$

In the initialization step, the assignment of colors needs to be justified locally and therefore we require that the colorings are *strictly* valid. Checking for validity takes $O(n_i^2)$ time; therefore, this step can be carried out in $O(3^{n_i} \cdot n_i^2)$ time.

Introduce Nodes

Let $X_i = \{x_{i_1}, \ldots, x_{i_{n_j}}, x\}$ be an introduce node with child node $X_j = \{x_{i_1}, \ldots, x_{i_{n_j}}\}$. Compute the table A_i as follows. For a coloring $c : X_i \to \{0, 1, 2\}$ and an index $1 \leq p \leq |X_i|$, define $\mathrm{gray}_p(c)$ to be a coloring derived from c by re-coloring the vertex with index p with color 2. Let $N_j(x)$ be the set of neighbors of vertex x in X_j, that is, $N_j(x) := N(x) \cap X_j$.

Then the mapping m_i in A_i is computed as follows (recall that m_i represents the number of *vertices* in an induced matching in the graph visited up to the current stage of the algorithm). For a coloring $c = (c_1, \ldots, c_{n_j})$ set

$$m_i(c \times \{0\}) := m_j(c). \tag{1}$$

$$m_i(c \times \{1\}) := \begin{cases} m_j(\mathrm{gray}_p(c)) + 1, & \text{if there is a vertex } x_{j_p} \in N_j(x) \\ & \text{with } c_p = 1, \text{ and for all} \\ & x_{j_q} \in N_j(x) \text{ with } q \neq p : c_q = 0. \\ -\infty, & \text{otherwise.} \end{cases} \tag{2}$$

$$m_i(c \times \{2\}) := \begin{cases} m_j(c) + 1, & \text{if } c_p = 0 \text{ for all } x_{j_p} \in N_j(x). \\ -\infty, & \text{otherwise.} \end{cases} \tag{3}$$

Assignment 1 is clearly correct, since the coloring $c \times \{0\}$ is valid for X_i if and only if c is valid for X_j. The value of m_i is the same for both colorings. If the newly introduced vertex x has color 1 (Assignment 2), then—since $c \times \{1\}$ must be valid—there must be a neighbor y with color 1 within the bag X_i; all the other neighbors of x in X_i must have color 0. This is insured by the assignment condition. To see the correctness of the computed value $m_i(c \times \{1\})$, note that y must have color 2 in bag X_j, since the partner of y was not yet known in the stage when the algorithm was processing bag X_j, and we increase the number of solution vertices by one since the newly introduced vertex has color 1. The condition of Assignment 3 simply verifies the validity of the coloring $c \times \{2\}$, and we increase the number of solution vertices by one since the newly introduced vertex has color 2.

For each row of table A_i, we have to look at the neighborhood of vertex x within the bag X_i to check whether the corresponding coloring is valid. Therefore, this step can be carried out in $O(3^{n_i} \cdot n_i)$ time.

Forget Nodes

Let $X_i = \{x_{i_1}, \ldots, x_{i_{n_i}}\}$ be a forget node with child node $X_j = \{x_{i_1}, \ldots, x_{i_{n_i}}, x\}$. Compute the table A_i as follows. For each coloring $c \in \{0, 1, 2\}^{n_i}$ set

$$m_i(c) := \max_{d \in \{0,1\}} \{m_j(c \times \{d\})\}.$$

The maximum is taken over colors 0 and 1 only, as a coloring $c \times \{2\}$ cannot be extended to a maximum induced matching. To see this, note that such a coloring assigns vertex x color 2 and since x is forgotten, by the consistency property of tree-decompositions (Property 3 of Definition 24), it does not appear in any of the bags that the algorithm sees in the future.

Clearly, this evaluation can be done in $O(3^{n_i} \cdot n_i)$ time. The crucial part are the join nodes.

Join Nodes

For a join node X_i with child nodes X_j and X_k compute the table A_i as follows. We say that two colorings $c' = (c'_1, \ldots, c'_{n_i}) \in \{0,1,2\}^{n_i}$ and $c'' = (c''_1, \ldots, c''_{n_i}) \in \{0,1,2\}^{n_i}$ are *correct* for a coloring $c = (c_1, \ldots, c_{n_i})$ if the following conditions hold for every $p \in \{1, \ldots, n_i\}$:

(1) if $c_p = 0$ then $c'_p = 0$ and $c''_p = 0$,

(2) if $c_p = 1$ then
  (a) if x_{i_p} has a neighbor $x_{i_q} \in X_i$ with $c_q = 1$ then $c'_p = c''_p = 1$,
  (b) else either $c'_p = 1$ and $c''_p = 2$, or $c'_p = 2$ and $c''_p = 1$, and

(3) if $c_p = 2$ then $c'_p = 2$ and $c''_p = 2$.

Then the mapping m_i of X_i is evaluated as follows. For each coloring $c \in \{0,1,2\}^{n_i}$ set

$$m_i(c) := \max\{m_j(c') + m_k(c'') - \#_1(c) - \#_2(c) \mid c' \text{ and } c'' \text{ are correct for } c\}.$$

In other words, we determine the value of $m_i(c)$ by looking up the corresponding coloring in m_j and in m_k (corresponding to the left and right subtree, respectively), add the corresponding values and subtract the number of vertices colored 1 or 2 by c, since they would be counted twice otherwise.

Clearly, if the coloring c assigns color 0 to a vertex $x \in X_i$, then so must colorings c' and c''. The same holds if c assigns color 2 to a vertex. However, if c assigns color 1 to a vertex x, then this coloring can be justified in two ways. The first case is when x has a neighbor $y \in X_i$ that is also colored 1. Then both colorings c' and c'' obviously assign 1 to x (and 1 to y). The second case is when all neighbors of x in X_i are assigned color 0. Then the assignment $c(x) = 1$ must be justified by another vertex in the solution which is in a bag which has already been processed in a previous stage of the algorithm. This vertex is located either in the left subtree or in the right subtree (corresponding to m_j or m_k, respectively), but not both. Therefore, the color of x can only be justified by assigning color 1 to x by c' and color 2 to x by c'', or vice versa.

Note that for a given coloring $c \in \{0,1,2\}^{n_i}$, with $a := \#_1(c)$, there are at most 2^a possible pairs of correct colorings for c. There are $2^{n_i-a}\binom{n_i}{a}$ possible colorings c with a vertices colored 1, thus

$$|\{(c', c'') \mid c \in \{0,1,2\}^{n_i}, c' \text{ and } c'' \text{ are correct for } c\}| \leq \sum_{a=0}^{n_i} 2^{n_i-a}\binom{n_i}{a} \cdot 2^a = 4^{n_i}.$$

Since we have to check the neighbors of x within X_i for each pair of correct colorings, the total running time for this step is $O(4^{n_i} \cdot n_i)$. In total, we get a running time of $O(4^{\omega} \cdot |I|)$ for the whole dynamic programming process. □

6 Conclusions and Outlook

As our main result, we have shown that Induced Matching on planar graphs admits a linear problem kernel. Additionally, we gave an improved dynamic programming algorithm for Induced Matching on graphs of bounded treewidth. The data reduction rules for the planar case are very simple and the kernelization can be done in linear time. The upper bound on the number of vertices inside regions can probably be improved using a more sophisticated analysis. More precisely, we feel that the approach used in Lemma 15 can be adapted and generalized to give a direct bound for the size of entire regions, and that a significant improvement of the constant in the kernel size is not too difficult to achieve. Note that with a different technique, a kernel of size $40k$ has recently been achieved [29]. It would be interesting to see whether the kernelization could be generalized to non-planar graphs such as in the case of Dominating Set [21]. Moreover, generalizing the data reduction rules could lead to an improved kernel (see, e.g., [2]). The properties of Induced Matching concerning approximation could be another interesting research field. Investigating the parameterized complexity of Induced Matching on other restricted classes of graphs may be of interest.

Acknowledgements

We thank Jiong Guo and Rolf Niedermeier (University of Jena, Germany) for initiating this research and for several constructive discussions and comments. We also thank Daniel Lokshtanov and Saket Saurabh (University of Bergen, Norway) for pointing out the $W[1]$-hardness of the Induced Matching problem on bipartite graphs. We also thank two anonymous referees for giving very helpful advice how to improve the presentation of this paper.

References

[1] J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002.

[2] J. Alber, B. Dorn, and R. Niedermeier. A general data reduction scheme for domination in graphs. In *Proceedings of SOFSEM'06*, volume 3831 of *LNCS*, pages 137–147. Springer, 2006.

[3] J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *Journal of the ACM*, 51(3):363–384, 2004.

[4] J. Alber and R. Niedermeier. Improved tree decomposition based algorithms for domination-like problems. In *Proceedings of LATIN'02*, volume 2286 of *LNCS*, pages 613–628. Springer, 2002.

[5] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.

[6] H. L. Bodlaender. Treewidth: Algorithmic techniques and results. In *Proceedings of MFCS'97*, volume 1295 of *LNCS*, pages 19–36. Springer, 1997.

[7] H. L. Bodlaender. Treewidth: Characterizations, applications, and computations. In *Proceedings of WG'06*, volume 4271 of *LNCS*, pages 1–14. Springer, 2006.

[8] K. Cameron. Induced matchings. *Discrete Applied Mathematics*, 24:97–102, 1989.

[9] K. Cameron. Induced matchings in intersection graphs. *Discrete Mathematics*, 278(1-3):1–9, 2004.

[10] K. Cameron, R. Sritharan, and Y. Tang. Finding a maximum induced matching in weakly chordal graphs. *Discrete Mathematics*, 266(1-3):133–142, 2003.

[11] K. Cameron and T. Walker. The graphs with maximum induced matching and maximum matching the same size. *Discrete Mathematics*, 299(1-3):49–55, 2005.

[12] J. Chen, H. Fernau, I. A. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. *SIAM Journal on Computing*, 37(4):1077–1106, 2007.

[13] M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems. In *Proceedings of ESA'04*, volume 3221 of *LNCS*, pages 192–203. Springer, 2004.

[14] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

[15] R. Downey, M. R. Fellows, and V. Raman. The complexity of irredundant sets parameterized by size. *Discrete Applied Mathematics*, 100(3):155–167, 2000.

[16] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[17] W. Duckworth, D. Manlove, and M. Zito. On the approximability of the maximum induced matching problem. *Journal of Discrete Algorithms*, 3(1):79–91, 2005.

[18] R. J. Faudree, E. Flandrin, and Z. Ryjácek. Claw-free graphs—a survey. *Discrete Mathematics*, 164(1-3):87–147, 1997.

[19] M. R. Fellows. The lost continent of polynomial time: Preprocessing and kernelization. In *Proceedings of IWPEC'06*, volume 4169 of *LNCS*, pages 276–277. Springer, 2006.

[20] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[21] F. V. Fomin and D. M. Thilikos. Fast parameterized algorithms for graphs on surfaces: Linear kernel and exponential speed-up. In *Proceedings of ICALP'04*, volume 3142 of *LNCS*, pages 581–592. Springer, 2004.

[22] G. Fricke and R. Laskar. String matching in trees. *Congressum Numerantium*, 89:239–243, 1992.

[23] M. C. Golumbic and R. Laskar. Irredundancy in circular arc graphs. *Discrete Applied Mathematics*, 44(1-3):79–89, 1993.

[24] M. C. Golumbic and M. Lewenstein. New results on induced matchings. *Discrete Applied Mathematics*, 101(1-3):157–165, 2000.

[25] Z. Gotthilf and M. Lewenstein. Tighter approximations for maximum induced matchings in regular graphs. In *Proceedings of WAOA'05*, volume 3879 of *LNCS*, pages 270–281. Springer, 2005.

[26] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007.

[27] J. Guo and R. Niedermeier. Linear problem kernels for NP-hard problems on planar graphs. In *Proceedings of ICALP'07*, volume 4596 of *LNCS*, pages 375–386. Springer, 2007.

[28] J. Guo, R. Niedermeier, and S. Wernicke. Fixed-parameter tractability results for full-degree spanning tree and its dual. In *Proceedings of IWPEC'06*, volume 4169 of *LNCS*, pages 203–214. Springer, 2006.

[29] I. A. Kanj, M. J. Pelsmajer, G. Xia, and M. Schaefer. On the induced matching problem. In *Proceedings of STACS'08*, pages 397–408. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2008.

[30] T. Kloks. *Treewidth, Computations and Approximations*, volume 842 of *LNCS*. Springer, 1994.

[31] J. Kneis, D. Mölle, S. Richter, and P. Rossmanith. Divide-and-color. In *Proceedings of WG'06*, volume 4271 of *LNCS*, pages 58–67. Springer, 2006.

[32] C. W. Ko and F. B. Shepherd. Bipartite domination and simultaneous matroid covers. *SIAM Journal on Discrete Mathematics*, 16(4):517–523, 2003.

[33] D. Kobler and U. Rotics. Finding maximum induced matchings in subclasses of claw-free and P_5-free graphs, and in graphs with matching and induced matching of equal maximum size. *Algorithmica*, 37(4):327–346, 2003.

[34] V. V. Lozin. On maximum induced matchings in bipartite graphs. *Information Processing Letters*, 81(1):7–11, 2002.

[35] V. V. Lozin and D. Rautenbach. Some results on graphs without long induced paths. *Information Processing Letters*, 88(4):167–171, 2003.

[36] H. Moser and D. M. Thilikos. Parameterized complexity of finding regular induced subgraphs. In *Proceedings of ACiD'06*, volume 7 of *Texts in Algorithmics*, pages 107–118. College Publications, 2006.

[37] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[38] Y. Orlovich, G. Finke, V. Gordon, and I. Zverovich. Approximability results for the maximum and minimum maximal induced matching problems. *Discrete Optimization*. To appear.

[39] E. Prieto and C. Sloper. Looking at the stars. *Theoretical Computer Science*, 351(3):437–445, 2006.

[40] V. Raman and S. Saurabh. Short cycles make W-hard problems hard: FPT algorithms for W-hard problems in graphs with no short cycles. *Algorithmica*. To appear.

[41] N. D. Roussopoulos. A $\max\{m, n\}$ algorithm for determining the graph H from its line graph G. *Information Processing Letters*, 2(4):108–112, 1973.

[42] L. J. Stockmeyer and V. V. Vazirani. NP-completeness of some generalizations of the maximum matching problem. *Information Processing Letters*, 15(1):14–19, 1982.

[43] M. Zito. Induced matchings in regular graphs and trees. In *Proceedings of WG'99*, volume 1665 of *LNCS*, pages 89–100. Springer, 1999.