The Parameterized Complexity of the Induced Matching Problem in Planar Graphs*

Hannes Moser[1]** and Somnath Sikdar[2]

[1] Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Germany
moser@minet.uni-jena.de

[2] The Institute of Mathematical Sciences,
C.I.T Campus, Taramani,
Chennai 600113, India
somnath@imsc.res.in

Abstract. Given a graph G and a nonnegative integer k, the NP-complete Induced Matching problem asks for an edge subset M 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. Our main contribution for the problem—which is W[1]-hard in general—is to show that it is fixed-parameter tractable on planar graphs by providing a linear problem kernel. Additionally, we generalize a known algorithm for Induced Matching on trees to graphs of bounded treewidth using an improved dynamic programming approach.

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 (1) M is a matching and (2) no two edges of M are joined by an edge of G. In other words, the subgraph induced by $V(M)$ is precisely the set M. The maximum size of an induced matching in G is denoted by $\mathrm{im}(G)$. The (decision version) of the Induced Matching problem asks the following question: given a graph G and an integer k, does G have an induced matching of size at least k? 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 [28] as the "risk-free" marriage problem[3]. This problem has been intensively studied in recent years. It is known to be NP-complete for planar graphs of maximum degree 4 [22], bipartite graphs of maximum degree 3, C_4-free bipartite graphs [24], r-regular graphs for $r \geq 5$, line-graphs and Hamiltonian graphs [23]. The problem is polynomial time solvable for trees [16,29], chordal graphs [7] and weakly chordal graphs [9]. Further results on special graph classes can be found in [8,17,18,23,24,25]. Regarding approximability, it is known that the Induced Matching problem is APX-hard on $4r$-regular graphs, for all $r \geq 1$ [29], and bipartite graphs with maximum degree 3 [12]. On the other hand, there exists an approximation algorithm with performance ratio $d - 1$ on d-regular graphs ($d \geq 3$) [12].

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 in the general case [26]. This result provides evidence that Induced Matching is not fixed-parameter tractable in general graphs. Therefore it is of interest to study the parameterized complexity of the problem in those restricted graph classes where it remains NP-complete. In this paper, we

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

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

[3] 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.

focus on planar graphs. The parameterized complexity of various NP-complete problems on planar graphs has already been studied. An interesting aspect of such studies are linear problem kernels. Kernelization is an important and powerful concept in Parameterized Complexity Theory used to demonstrate fixed-parameter tractability. One can consult the recent surveys by Fellows [13], Guo and Niedermeier [19], and the book by Flum and Grohe [14] for an overview on kernelization. Dominating Set was one of the first problems for which a linear kernel on planar graphs was found [2]. The kernel developed in [2] has subsequently been improved in [10]. In the same paper, the authors describe lower bound results on the kernel size for several problems on planar graphs including Dominating Set and Vertex Cover. The technique developed in [2] has been exploited in developing a linear kernel for the Full-Degree Spanning Tree [20], a maximization problem, and has also been extended to graphs of bounded genus [15]. Thus far, the technique has been applied to problems whose solutions are vertex subsets. We give the first application of this technique for a maximization problem whose solutions are edge subsets.

We show that Induced Matching on planar graphs admits a linear-size problem kernel. We adapt and extend the technique introduced in [2] and [20]. The corresponding data reduction rules can be carried out in linear time. The organization of the remaining paper is as follows. In Section 2, we start out with some basic definitions and notation used in the rest of the paper. In Section 3, we first state the data reduction rules that we need and then present the kernelization proof, which is also the main technical contribution of this paper. Additionally, in Section 4, we generalize an algorithm for Induced Matching on trees [29] to graphs of bounded treewidth using an improved dynamic programming approach.

2 Preliminaries

In this paper, we deal with fixed-parameter algorithms that emerge from the field of parameterized complexity analysis [11,27], 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 a another instance (I', k') in polynomial time, such that (I, k) is a yes-instance iff (I', k') is a yes-instance. 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. Analogously to classical complexity theory, Downey and Fellows [11] 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 [11].

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, we use n to denote the number of vertices of a given graph. 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 $G[V \setminus V']$. If $v \in V$ we also write $G - v$ instead of $G \setminus \{v\}$. The *(open) neighborhood* $N(v)$ of a vertex v is the set of all vertices in $V - v$ that are adjacent to 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 in 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 in C.

3 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 [2,10,20], these data reduction rules are quite simple and therefore can be carried out much more efficiently.

Degree Zero Rule Delete vertices of degree zero.
Degree One Rule If a vertex u has two distinct neighbors x, y of degree 1, then delete x.
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 $\deg(u) = \deg(v) = 2$, then delete x.

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

Lemma 1. *The three rules are correct and 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.*

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

Theorem 1. *Let $G = (V, E)$ be a planar graph reduced with respect to the Degree Zero, the Degree One, and the Degree Two Rules. 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 $u \in V(M)$ such that $d(u, v) \leq 2$. For 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" the matching edges that are close to each other. However, there can be many vertices in the vicinity of a vertex in M. 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 appears in [2], but the definition of a region as it appears there is much simpler as the distance between any two vertices of a minimum dominating set is at most three. Not only is the definition of a region more involved in our case, the proof of a bounded number of vertices inside and outside of regions is more complicated too. The remaining part of this section is dedicated to the proof of Theorem 1. First, in Section 3.1 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 3.2, we show that a region in such a maximal region decomposition contains only a constant number of vertices. Finally, in Section 3.3 we show that in any reduced graph there are only $O(|M|)$ vertices which are not contained in regions.

3.1 Finding a Maximal Region Decomposition

Definition 1. *Let G be a plane graph and M a maximum induced matching of G. For two 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 $a_2 \in e_2$, and*
   - *$b_1, \ldots, b_2$, $b_1 \neq b_2$, between $b_1 \in e_1$ and $b_2 \in e_2$, and*
   - *by the edge e_1 if $a_1 \neq b_1$, and*
   - *by the edge e_2 if $a_2 \neq b_2$, and*
2. *for each vertex x in the region $R(e_1, e_2)$ there exists $y \in V(\{e_1, e_2\})$ 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.*

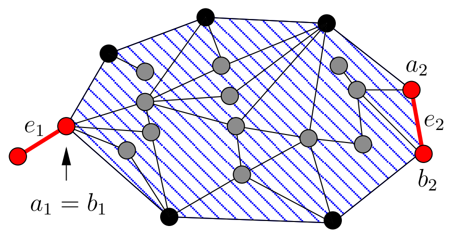

Fig. 1. Example of region $R(e_1, e_2)$ between two edges $e_1, e_2 \in M$. Note that e_1 is not part of R, but only its endpoint $a_1 = b_1$. The black vertices are the boundary vertices, and the gray vertices in the hatched area are the vertices strictly inside of R.

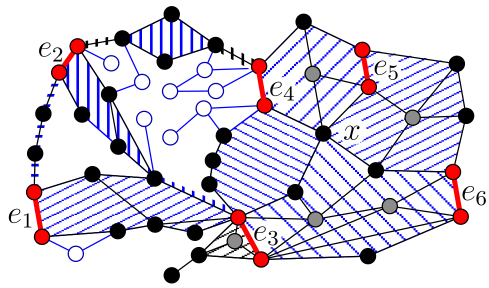

Fig. 2. Example of an M-region decomposition. The black vertices are boundary vertices, the gray vertices are strictly inside of a region and the vertices depicted as small circles are outside of regions. Each region is hatched with a pattern. Note the special cases, as for instance regions that consist 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.

The set of boundary vertices of R is denoted by δR. Vertices of $R(e_1, e_2)$ that are not in δR are ***strictly inside*** *R.* ***An edge*** *$\{u, v\}$* ***is strictly inside*** *R if both u and v are strictly inside R.* ***A path*** *P* ***lies strictly inside*** *R if all vertices on that path lie strictly inside R. 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 1.

Definition 2. *An* ***M-region decomposition*** *of $G = (V, E)$ is a set of regions $\mathcal{R}$ 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 $V(\mathcal{R}) := \bigcup_{R \in \mathcal{R}} V(R)$. We call an M-region decomposition* ***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 Figure 2.

Lemma 2. *Given a plane reduced graph $G = (V, E)$ and a maximum induced matching M of G, there exists an algorithm that constructs a maximal M-region decomposition with $O(|M|)$ regions.*

This lemma can be proved by exhibiting a greedy algorithm that builds a maximal M-region decomposition in a stepwise manner by searching a region of maximal size that is not yet in the region decomposition at the actual step of the algorithm. Since this approach is similar to the algorithms by Alber et al. [2] and Guo et al. [20] we omit the details here.

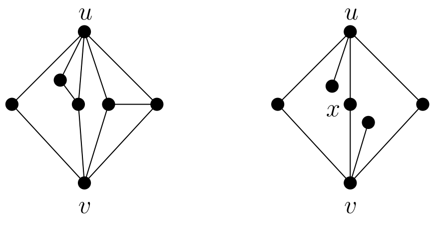

Fig. 3. A diamond (left) and an empty diamond (right) in a reduced plane graph.

3.2 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 to 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 to 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. We call such an area a *diamond*.

Definition 3. *Let u and v be two vertices in a plane graph. A* ***diamond*** *is a 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.*

See Figure 3 for an example of an empty and a non-empty diamond. In a reduced plane graph empty diamonds have a very restricted size. We are especially interested in the maximum number of vertices strictly inside an empty diamond that have both u and v as neighbors. The following lemma is easy to show.

Lemma 3. *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.*

This lemma shows that if there are more than three 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 plane reduced graph is bounded.

Lemma 4. *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 for 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:*

If neither u nor v is an endpoint of an edge in M, then there are at most 5 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 always exhibit an induced matching M' with $|M'| > |M|$, a contradiction.

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 Figure 4 (left-hand side), with w_1 and w_6 lying on the enclosing paths. Consider the diamond with the boundary induced by the vertices $u, v, w_2, \ldots, w_5$. Since there is more than one vertex with neighbors u and v strictly inside that diamond (namely, w_3, w_4), we know that it is not empty due to Lemma 3; that is, there exists an edge e in this diamond which is not incident to u or v. Clearly e is not incident to either w_1 or w_6 and so e is not incident to any

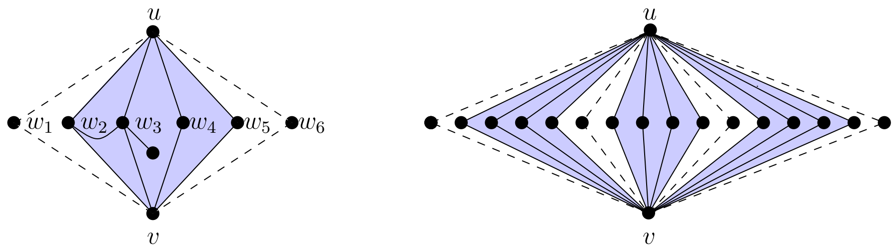

Fig. 4. Embedding of the vertices $w_1, \ldots, w_6$ for the first case in the proof of Lemma 4 (left), and an embedding of 16 neighbors of u and v for the last case of the proof, where possible vertices and edges not on the considered paths are not drawn (right). The diamonds are shaded, and the "isolation paths" are drawn with dashed lines.

vertex in $V(M)$. Therefore, we can add e to the induced matching M, contradicting its maximum cardinality.

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 we can see that assuming 11 length-2 paths between u and v (using $(u, w_1, v), (u, w_6, v)$, and (u, w_{11}, v) as "isolation paths") leads to at least two non-empty diamonds whose boundaries share only u and v, thus we can replace the edge e in M by two edges (one from each non-empty diamond) which contradicts the maximum cardinality of M.

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 4). 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 4 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. The first lemma (Lemma 5) upper-bounds the number of such vertices under the condition that they are contained in an area which is enclosed by a short cycle. This lemma is then used in Lemma 6 to upper-bound the total number of such vertices for a given boundary vertex.

Lemma 5. *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 of 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. ☐

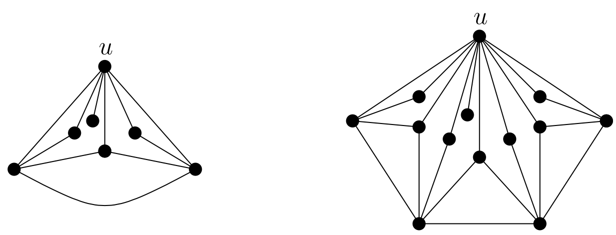

Fig. 5. Worst-case embeddings to illustrate Lemma 5: There are four (left-hand side) and eight (right-hand side) neighbors of u strictly inside of A.

See Figure 5 for two examples for different lengths of P that contain a maximum number of neighbors of u strictly inside A such there is no edge strictly inside A.

Lemma 6. *Let u be a boundary vertex of a region $R(e_1, e_2)$ in a reduced plane graph G, and let M be a maximum induced matching for G. If u has at least 41 neighbors $v_1, \ldots, v_{41}$ strictly inside R that are not adjacent to any other boundary vertex, then we can find an induced matching M' with $|M'| > |M|$.*

Proof. 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 5, 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 4 and Lemma 6, we can now upper-bound the number of vertices inside a region.

Lemma 7. *A 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 A and B, where A contains all vertices at distance exactly one to some boundary vertex, and B contains 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 6, a vertex $v \in \delta R$ on the boundary can have at most 41 neighbors in A_1. Since a region has at most ten boundary vertices, we conclude that A_1 contains at most 410 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. After this, merge multiple edges between two boundary vertices into one. 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. Due to Lemma 4, 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 an graph without edges (due to the maximum cardinality of M). 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 vertices in B with degree at least two, we use the same argument as that used to the bound of the size of A_2. Since $|A| = O(1)$, there are a constant number of degree-at-least-two vertices in B. Therefore $|B| = O(1)$. This completes the proof.

□

With this lemma, we can easily see that there is only a linear number of vertices contained in regions.

Proposition 1. *Let G be a reduced plane graph and let M be a maximum induced matching for G. There exists an M-region decomposition such that the total number of vertices inside all regions is $O(|M|)$.*

Proof. Due to Lemma 2 we can find a maximal M-region decomposition for G with at most $O(|M|)$ regions. Inside of each of the $O(|M|)$ regions there is only a constant number of vertices due to Lemma 7. □

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

3.3 Bounding the Number of Vertices Lying Outside Regions

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

Note again that the distance from any vertex to a vertex in $V(M)$ is at most two, as otherwise we would contradict the maximum cardinality of M. We subdivide the set of vertices outside regions into two disjoint subsets A and B, where

- A is the set of vertices at distance exactly one to some vertex in $V(M)$,
- B is the set of vertices at distance at least two from every vertex in $V(M)$.

We bound the size 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 another vertex $w \in V(M)$, 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 8. *Let v be a vertex in A_1 and let u be its neighbor in $V(M)$. Then for all $w \in V(M) - u$, $d(v, w) \geq 3$.*

Proof. Let u and v be as in the statement of the Lemma and let $w \in V(M)\ \{u\}$. 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 9. *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. To show this claim, we use a similar argument like in Lemma 6. Suppose for the purpose of contradiction that 21 vertices $\{v_1, \dots, v_{21}\} \subseteq A_1$ adjacent to u are embedded in a clockwise fashion around u. Let e be the edge in M which is incident to u. First, suppose that v_1 and v_{11} have distance at least four in $G - u$. Then, two edges e_a, e_b in $G - u$ incident to v_1 and v_{11}, respectively, are pairwise non-adjacent. Moreover, they are not adjacent to any vertex in $V(M)$ in $G - u$ due to Lemma 8. Therefore, we obtain a induced matching $M' = (M - e) \cup \{e_1, e_b\}$ of larger size than M, contradicting its maximum cardinality. The same holds if the distance between v_{11} and v_{21} is at least four in $G - u$. Thus, the last case to consider is that v_1 and v_{11} have distance at most three and that v_{11} and v_{21} have distance at most three. Let P_1 and P_2 be two shortest paths between v_1 and v_{11}, and between v_{11} and v_{21}, respectively. Note that due to Lemma 8 the two paths cannot contain any vertex from $V(M)$. For the same reason as in Lemma 6 the two paths enclose two areas, each of which contains an edge strictly inside due to Lemma 5. The edge e can be replaced by these two edges, contradicting the maximum cardinality of M. This shows the above claim. Since there are exactly $2\,|M|$ vertices in $V(M)$, the claim shows that the total number of vertices in A_1 is at most $40\,|M|$.

The size of A_2 can be bounded as follows. Every vertex v in A_2 is adjacent to a vertex in $u \in V(M)$ and some other boundary vertex w. This boundary vertex w must be adjacent to u, since otherwise there is a path consisting of the subpath (u, v, w) and some subpath on the boundary on which w lies, and this path could be added to the region decomposition, contradicting its maximality. Since there are $O(|M|)$ regions, there are $O(|M|)$ possible boundary vertices adjacent to a vertex in $V(M)$. Due to Lemma 4 there can be at most 10 vertices adjacent to a vertex in $V(M)$ that are adjacent to the same boundary vertex, thus we obtain that A_2 contains $O(|M|)$ vertices. □

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

Lemma 10. *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 9 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 that in Lemma 7 (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 2. *Given a maximal M-region decomposition with $O(|M|)$ regions, the number of vertices that lie outside of regions is $O(|M|)$.*

Proof. Follows from Lemmas 9 and 10. □

Using Propositions 1 and 2, we eventually can show that for a given graph G and a maximum induced matching M there exists a 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 1, that is, Maximum Induced Matching admits a linear problem kernel on planar graphs.

4 Induced Matching on Graphs with Bounded Treewidth

Zito [29] developed a linear-time dynamic programming algorithm to solve Induced Matching on trees. We generalize this approach to obtain a linear-time algorithm on graphs of bounded treewidth. 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 [3]. Here we describe only those parts of the algorithm which are important in showing the improved running time. The basic definitions and the dynamic programming technique itself will not be explained here due to space restrictions. We refer the reader to the standard literature about tree decompositions [4,5,6,21].

Theorem 2. *Let $G = (V, E)$ be a graph with a given nice tree decomposition $(\{X_i \mid i \in I\}, T)$. The size of a maximum induced matching of G can be computed in $O(4^\omega \cdot |I|)$ time, where 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 $G[X_i]$. 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 labelled 0 is not contained in M.
black(1): A vertex labelled 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 labelled 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 $n_i + 1$st 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. 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 labelled 1 and 2 has the following structure: vertices labelled 2 have degree 0 and those labelled 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 labelled 1 induce isolated edges. We next describe the dynamic programming process. Recall that we assume that we work with a nice tree decomposition.

The dynamic programming for the leaf, introduce, and forget nodes is relatively straight-forward and is therefore moved to the appendix. The tables for each of these nodes can be computed in $O(3^{n_i} \cdot n_i)$ time. The crucial part are the 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. □

5 Conclusions and Outlook

As our main result, we have shown that Planar Induced Matching admits a linear problem kernel. Additionally, we gave an improved dynamic programming algorithm for that problem 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 5 can be adapted and generalized to give a direct bound for the size of entire regions.

It would be interesting to see whether the kernelization could be generalized to non-planar graphs such as in the case of Dominating Set [15]. Another possible research topic could be search tree algorithms for planar graphs. For Dominating Set on planar graphs, there exists a search tree algorithm [1], and it is open whether a similar result for Induced Matching on planar graphs is possible. 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. We can show simple problem kernels for bounded-degree graphs, graphs of girth at least 6, C_4-free bipartite graphs, and line graphs. A class of major interest are bipartite graphs, where the parameterized complexity of Induced Matching is open.

6 Acknowledgements

We thank Jiong Guo and Rolf Niedermeier (University of Jena, Germany) for initiating this research and for several constructive discussions and comments.

References

1. J. Alber, H. Fan, M. R. Fellows, H. Fernau, R. Niedermeier, F. A. Rosamond, and U. Stege. A refined search tree technique for dominating set on planar graphs. *Journal of Computer and System Sciences*, 71(4):385–405, 2005.
2. J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *Journal of the ACM*, 51(3):363–384, 2004.
3. J. Alber and R. Niedermeier. Improved tree decomposition based algorithms for domination-like problems. In *LATIN 2002: Theoretical Informatics, 5th Latin American Symposium*, volume 2286 of *LNCS*, pages 613–628. Springer, 2002.

4. H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.
5. H. L. Bodlaender. Treewidth: Algorithmoc techniques and results. In *Mathematical Foundations of Computer Science 1997: 22nd International Symposium (MFCS'97)*, volume 1295 of *LNCS*, pages 19–36. Springer, 1997.
6. H. L. Bodlaender. Discovering treewidth. In *SOFSEM 2005: Theory and Practice of Computer Science, 31st Conference on Current Trends in Theory and Practice of Computer Science*, volume 3381 of *LNCS*, pages 1–16. Springer, 2005.
7. K. Cameron. Induced matchings. *Discrete Applied Mathematics*, 24:97–102, 1989.
8. K. Cameron. Induced matchings in intersection graphs. *Discrete Mathematics*, 278(1-3):1–9, 2004.
9. K. Cameron, R. Sritharan, and Y. Tang. Finding a maximum induced matching in weakly chordal graphs. *Discrete Mathematics*, 266(1-3):133–142, 2003.
10. J. Chen, H. Fernau, I. A. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. In *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS'05)*, pages 269–280, 2005.
11. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.
12. 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.
13. M. R. Fellows. The lost continent of polynomial time: Preprocessing and kernelization. In *Proceedings of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC'06)*, volume 4169 of *LNCS*, pages 276–277. Springer, 2006.
14. J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, 2006.
15. F. V. Fomin and D. M. Thilikos. Fast parameterized algorithms for graphs on surfaces: Linear kernel and exponential speed-up. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, volume 3142 of *LNCS*, pages 581–592. Springer, 2004.
16. G. Fricke and R. Laskar. String matching in trees. *Congressum Numerantium*, 89:239–243, 1992.
17. M. C. Golumbic and R. Laskar. Irredundancy in circular arc graphs. *Discrete Applied Mathematics*, 44(1-3):79–89, 1993.
18. M. C. Golumbic and M. Lewenstein. New results on induced matchings. *Discrete Applied Mathematics*, 101(1-3):157–165, 2000.
19. J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News.* To appear, 2007.
20. J. Guo, R. Niedermeier, and S. Wernicke. Fixed-parameter tractability results for full-degree spanning tree and its dual. In *Proceedings of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC'06)*, volume 4169 of *LNCS*, pages 203–214. Springer, 2006.
21. T. Kloks. *Treewidth, Computation and Approximation*, volume 842 of *LNCS.* Springer, 1994.
22. C. W. Ko and F. B. Shepherd. Bipartite domination and simultaneous matroid covers. *SIAM Journal on Discrete Mathematics*, 16(4):517–523, 2003.
23. 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.
24. V. V. Lozin. On maximum induced matchings in bipartite graphs. *Information Processing Letters*, 81(1):7–11, 2002.
25. V. V. Lozin and D. Rautenbach. Some results on graphs without long induced paths. *Information Processing Letters*, 88(4):167–171, 2003.
26. H. Moser and D. M. Thilikos. Parameterized complexity of finding regular induced subgraphs. In *Proceedings of the 2nd Algorithms and Complexity in Durham Workshop (ACiD'06)*, volume 7 of *Texts in Algorithmics*, pages 107–118. College Publications, 2006.
27. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.
28. 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.
29. M. Zito. Induced matchings in regular graphs and trees. In *Proceedings of the 25th International Graph Theoretic Concepts in Computer Science Workshop (WG'99)*, volume 1665 of *LNCS*, pages 89–100. Springer, 1999.

APPENDIX

Dynamic Programming - Leaf Nodes, Introduce Nodes, and Forget Nodes

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 be *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 $\text{gray}_p(c)$ 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 map m_i in A_i is computed as follows. For each coloring $c = (c_1, \ldots, c_{n_j}) \in \{0, 1, 2\}^{n_j}$ set

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

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

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

Assignment 1 is clearly correct, since the coloring $c \times \{0\}$ is valid for X_i iff 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. 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, 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.