LCNS, Vol 1851, pp. 97–110, 2000

Fixed Parameter Algorithms for Planar Dominating Set and Related Problems

Jochen Alber*[1], Hans L. Bodlaender[2], Henning Fernau[1], and Rolf Niedermeier[1]

[1] Universität Tübingen, WSI für Informatik, Sand 13,
72076 Tübingen, Fed. Rep. of Germany,
{alber,fernau,niedermr}@informatik.uni-tuebingen.de
[2] Utrecht University, Department of Computer Science,
Utrecht, The Netherlands,
hansb@cs.uu.nl

Abstract. We present an algorithm for computing the domination number of a planar graph that uses $O(c^{\sqrt{k}}n)$ time, where k is the domination number of the given planar input graph and $c = 3^{6\sqrt{34}}$. To obtain this result, we show that the treewidth of a planar graph with domination number k is $O(\sqrt{k})$, and that such a tree decomposition can be found in $O(\sqrt{k}n)$ time. The same technique can be used to show that the DISK DIMENSION problem (find a minimum set of faces that cover all vertices of a given plane graph) can be solved in $O(c_1^{\sqrt{k}}n)$ time for $c_1 = 2^{6\sqrt{34}}$. Similar results can be obtained for some variants of DOMINATING SET, e.g., INDEPENDENT DOMINATING SET.

1 Introduction

A *k-dominating set D* of an undirected graph G is a set of k vertices of G such that each of the rest of the vertices has at least one neighbor in D. A minimal k such that the graph G has a k-dominating set is called the *domination number* of G.

The k-DOMINATING SET problem, i.e., the task to decide, given a graph $G = (V, E)$ and a positive integer k, whether or not there exists a k-dominating set, is among the core problems in algorithms, combinatorial optimization, and computational complexity [11, 16, 19, 24]. The problem is NP-complete, even when restricted to planar graphs with maximum vertex degree 3 and to planar graphs that are regular of degree 4 [16].

The approximability of the DOMINATING SET problem has received considerable attention [11, 19]. It is not known and is not believed that DOMINATING SET for general graphs has a constant factor approximation algorithm (see Crescenzi and Kann [11] for details). However, the PLANAR DOMINATING SET problem (i.e., the dominating set problem restricted to planar graphs) possesses a polynomial

* Work supported by the DFG-research project PEAL (Parameterized complexity and Exact ALgorithms), NI 369/1-1.

time approximation scheme [1]. That is, there is a polynomial time approximation algorithm with approximation factor $1+\epsilon$, where ϵ is a constant arbitrarily close to 0. However, the degree of the polynomial grows with $1/\epsilon$. Hence, applying the approximation scheme does not always lead to practical solutions and finding an "efficient" exact algorithm for PLANAR DOMINATING SET is therefore of interest.

Due to the hardness and relevance of DOMINATING SET, numerous papers have studied special cases of DOMINATING SET, e.g., connected dominating set, total dominating set, independent dominating set, dominating clique, and/or the complexity of the problem in special graph classes [6, 8, 10, 15, 17, 18, 22, 26]. For example, a very recent result shows that there is a factor $2+\epsilon$ approximation algorithm for DOMINATING SET on the class of circle graphs [12].

Lately, it has become popular to cope with computational intractability in a different way besides approximation: parameterized complexity [14]. Here, the basic observation is that, for many hard problems, the seemingly inherent combinatorial explosion can be restricted to a "small part" of the input, the *parameter.* For instance, the VERTEX COVER problem can be solved by an algorithm with running time $O(kn + 1.3^k)$ [9, 23], where the parameter k is a bound on the maximum size of the vertex cover set we are looking for and n is the number of vertices in the given graph. The fundamental assumption is $k \ll n$. As can easily be seen, this yields an efficient, practical algorithm for small values of k. A problem is called *fixed parameter tractable* if it can be solved in time $f(k)n^{O(1)}$ for an arbitrary function f which depends only on k. Unfortunately, according to the theory of parameterized complexity it is very unlikely that the DOMINATING SET problem is fixed parameter tractable. On the contrary, it was proven to be complete for $W[2]$, a "complexity class of parameterized intractability" (refer to Downey and Fellows [14] for any details). However, PLANAR k-DOMINATING SET is fixed parameter tractable. Downey and Fellows [13, 14] state an $O(11^k n)$ time bound for this problem, where n is the number of vertices.

In this paper, we present a drastic asymptotic improvement of this result. We show that PLANAR DOMINATING SET can be solved in time $O(c^{\sqrt{k}}n)$ for some constant c. To the best of our knowledge, this is the first fixed parameter tractability result where the exponent of the exponential term is not growing linearly, but with the square root of the parameter. We show that a graph with a dominating set of size k has treewidth $O(\sqrt{k})$, and we use this to solve PLANAR DOMINATING SET using the corresponding tree decomposition of the graph. Unfortunately, the constant base c of the exponential term that appears in the running time of our algorithm still is quite large, namely $c = 3^{6\sqrt{34}}$. However, the authors are confident that a more refined analysis of the applied techniques can improve this constant considerably.

Our technique can also be used to significantly improve a known bound for the DISK DIMENSION problem [2, 25]. The problem is defined as follows [2, 25]: Given a plane graph G, i.e., a graph with a fixed embedding in the plane and a positive integer k, is there a set of at most k faces (disks), such that all of the graph vertices are covered? The problem is NP-complete [2]. Downey and

Fellows [14] gave an $O(12^k n)$ algorithm for this problem. For a slightly more general version of the problem, Bienstock and Monma [2] showed that there is a time $O(c^k n)$ algorithm, where c is an unspecified constant. In this paper, we give an algorithm that solves DISK DIMENSION in time $O(c_1^{\sqrt{k}} n)$ for some constant c_1. We also discuss some variants of the DOMINATING SET problem.

2 Preliminaries

In this section, we provide necessary notions and some known results. We assume familiarity with basic graph-theoretical notation.

Definition 1 A graph G is *outerplanar* if there is a crossing-free embedding of G in the plane such that all vertices are on the same face.

Definition 2 A graph G is *r-outerplanar* if, for $r = 1$, G is outerplanar or, for $r > 1$, G has a planar embedding such that if all vertices on the exterior face (which form the *exterior layer* L_1) are deleted, the connected components of the remaining graph are all at most $(r-1)$-outerplanar.

In this way, we may speak of the layers $L_1, \ldots, L_r$ of an r-outerplanar graph. One easily makes the following central observation:

Proposition 1. *If a planar graph $G = (V, E)$ has a k-dominating set, then it can be at most $3k$-outerplanar.*

The main tool we use in our algorithm is a suitable tree decomposition:

Definition 3 Let $G = (V, E)$ be a graph. A *tree decomposition* of G is a pair $\langle \{X_i \mid i \in I\}, T \rangle$, where each X_i is a subset of V and T is a tree with the elements of I as nodes. The following three properties should hold:

- $\bigcup_{i \in I} X_i = V$;
- for every edge $\{u, v\} \in E$, there is an $i \in I$ such that $\{u, v\} \subseteq X_i$;
- for all $i, j, k \in I$, if j lies on the path between i and k in T, then $X_i \cap X_k \subseteq X_j$.

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

In [21, Table 2, page 550] or [5, Theorem 83], we can find:

Proposition 2. *An r-outerplanar graph has treewidth of at most $3r - 1$.*

Propositions 1 and 2 imply that a graph with domination number k has bounded treewidth, or, more precisely, its treewidth is bounded by $9k - 1$, but we will give a stronger bound later.

Theorem 4. *If a tree decomposition of width at most ℓ of a graph is known, then a minimum dominating set can be determined in time $3^\ell n$, where n is the number of nodes of the tree decomposition.*

Comments on the proof: The theorem can be proved by using dynamic programming techniques. For each bag (i.e., each set X_i of the corresponding tree decomposition) one keeps a table. These tables store, for every vertex in the bag, the information of whether that vertex is assumed to belong to either the dominating set, the (known) set of dominated vertices, or the set of vertices whose status is unknown at the given point. Since $|X_i| \leq \ell$, the table size for each bag is bounded by 3^ℓ. See e.g., [4, 27].

In this way, a straightforward solution to the PLANAR DOMINATING SET problem using tree decompositions leads to an algorithm which runs in time $O(3^{9k}n)$. (For a graph $G = (V, E)$, there always is a tree decomposition with optimal width and with at most $|V|$ nodes.) Downey and Fellows [13, 14] suggested an idea that leads to a faster search tree algorithm. They state an algorithm with running time $O(11^k n)$ (without using tree decompositions).

In what follows, we show that a graph with a k-dominating set has treewidth $O(\sqrt{k})$. Combining this with Theorem 4 gives a significant asymptotic improvement of the result of Downey and Fellows.

To understand the following technique, it is helpful to consider the concept of a *layer decomposition* of an r-outerplanar graph G. It is a forest of height r which is defined as follows: the nodes of the trees are sets of vertices of G and the different trees correspond to different components of G. In general, the ith layer of the layer decomposition forest defines a set of vertices L_i, namely the ith layer of G.

Consider now the ith layer of the forest, i.e., the nodes of level i in the decomposition forest, consisting of, possibly, several vertex sets $C_{i,1}, \ldots, C_{i,\ell_i}$. In other words, $L_i = \bigcup_{j=1}^{\ell_i} C_{i,j}$. The vertex sets $C_{i,1}, \ldots, C_{i,\ell_i}$ correspond to the vertices of different components of the subgraph induced by L_i. We refer to $C_{i,j}$ as a *layer-component*. In particular, the first layer consists of layer-components each of which equals the vertices from L_1 of one particular component.

A layer-component $C_{i,j}$ of layer L_i is called *non-empty* if it contains vertices from layer L_{i+1} in its interior.

Definition 5 Let $\emptyset \neq C \subseteq C_{i,j}$ be a subset of a non-empty layer-component $C_{i,j}$ of layer i, where $i \geq 2$. Then the unique cycle $B(C)$ in layer L_{i-1}, such that C is contained in the region enclosed by $B(C)$ and no other vertex of layer L_{i-1} is contained in this region, is called the *boundary cycle* of C.

The existence and uniqueness of such a boundary cycle $B(C)$ is easy to see.

3 Domination versus treewidth

Our algorithm is based on Theorem 4. Therefore, in the following we show that a planar graph with domination number k has treewidth of at most $O(f(k))$, where $f(k)$ is a sublinear function, which we are going to determine. Here, the main idea is to find small separators of the graph and merge the tree decompositions of the resulting subgraphs. To this end, the following observation is used.

Proposition 3. *If a connected graph can be decomposed into components of treewidth of at most t by means of a separator of size s, then the whole graph has treewidth of at most $t+s$.*

The proof is quite simple: Just merge the separator to every node in each tree decomposition of width at most t which correspond to the distinct components. Then add some arbitrary connections between the trees corresponding to the components in order to form a tree decomposition of the whole graph.

For planar graphs, there is an iterated version of this observation.

Proposition 4. *Let G be a planar graph with layers L_i, $(i = 1, \dots, r)$. For $i = 1, \dots, \ell$, let $\mathcal{L}_i$ be a set of consecutive layers, i.e. $\mathcal{L}_i = \{L_{j_i}, L_{j_i+1}, \dots, L_{j_i+n_i}\}$, such that $\mathcal{L}_i \cap \mathcal{L}_{i'} = \emptyset$ for all $i \neq i'$. Moreover, suppose G can be decomposed into components, each of treewidth of at most t, by means of separators $S_1, \dots, S_\ell$, where $S_i \subseteq \bigcup_{L \in \mathcal{L}_i} L$ for all $i = 1, \dots, \ell$. Then G has treewidth of at most $t + 2s$, where $s = \max_{i=1,\dots,\ell} |S_i|$.*

The proof again uses the merging-techniques illustrated in the previous proposition: Suppose, w.l.o.g., the sets $\mathcal{L}_i$ appear in successive order, i.e. $j_i < j_{i+1}$. For each $i = 0, \dots, \ell$, consider the component G_i of treewidth at most t which is cut out by the separators S_i and S_{i+1} (by default we set $S_0 = S_{\ell+1} = \emptyset$). We add S_i and S_{i+1} to every node in a given tree decomposition of G_i. In order to obtain a tree decomposition of G, we successively add an arbitrary connection between the trees T_i and T_{i+1} of the so-modified tree decompositions that correspond to the subgraphs G_i and G_{i+1}.

Finally, we still have to show how to construct (in polynomial time) a tree decomposition of width $f(k)$ matching our theoretical treewidth bound. This allows us to apply Theorem 4 to actually determine the dominating set we are aiming at.

The whole algorithm we present has time complexity $O(3^{f(k)}n)$. Since $f(k) \in O(\sqrt{k})$, this obviously gives an asymptotic improvement of the $O(11^k n)$ algorithm presented by Downey and Fellows.

In the following, we assume that our graph has a fixed plane embedding with r layers. We show that the treewidth cannot exceed $f(k)$ if a dominating set of size k is given.

3.1 Separators and treewidth

We assume that we have a dominating set D of size at most k. Let t_i be the number of vertices of $D_i = D \cap L_i$. Hence, $\sum_{i=1}^{r} t_i = k$. In order to avoid case distinctions, we set $t_0 = t_{r+1} = t_{r+2} = 0$. Moreover, let c_i denote the number of non-empty layer-components of layer L_i.

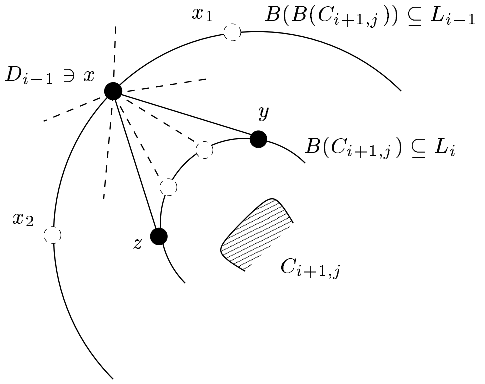


Fig. 1. upper triples

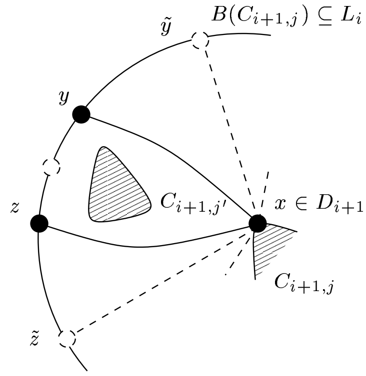
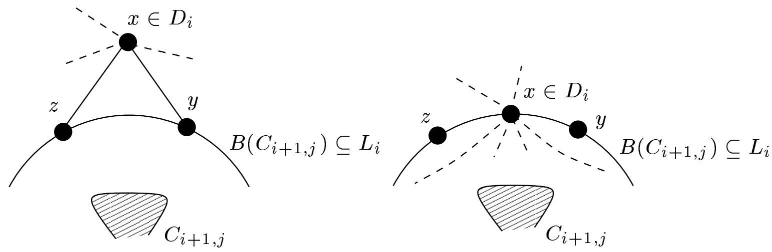


Fig. 2. lower triples



Case 1: Case 2:

Fig. 3. middle triples

We need some definitions for certain triples in the plane graph. These triples are defined in a way such that the union of these triples will yield separators of small size.

We define the triples for a layer L_i. The union of these triples separates vertices of layer L_{i-1} from vertices of layer L_{i+2}. For this purpose, in the following, we write $N(x)$ to describe the set of neighbors of a vertex x and use the notion $B(\cdot)$ for boundary cycles as introduced in Definition 5.

Definition 6 An *upper triple* for layer L_i is associated to a non-empty layer-component $C_{i+1,j}$ of layer L_{i+1} and a vertex $x \in D_{i-1}$ that has a neighbor on the boundary cycle $B(C_{i+1,j})$ (see Fig. 1). Then, clearly, $x \in B(B(C_{i+1,j}))$, by definition of a boundary cycle. Let x_1 and x_2 be the neighbors of x on the cycle $B(B(C_{i+1,j}))$. Starting from x_1, we go around x up to x_2 so that we visit all neighbors of x in layer L_i. We note the neighbors of x on the boundary cycle $B(C_{i+1,j})$. Going around gives two outermost neighbors y and z on this boundary cycle. The triple then is the three-element set $\{x, y, z\}$. In case x has only a single neighbor y in $B(C_{i+1,j})$, the "triple" consists of only $\{x, y\}$.

For each non-empty layer-component $C_{i+1,j}$ of L_{i+1} and each vertex $x \in D_{i-1}$ with neighbors in $B(C_{i+1,j})$, we obtain such an upper triple.

Definition 7 A *lower triple* for layer L_i is associated to a vertex $x \in D_{i+1}$ and a non-empty layer-component $C_{i+1,j'}$ of layer L_{i+1} (see Fig. 2). Suppose x lies in layer-component $C_{i+1,j}$. We only consider layer-components $C_{i+1,j'}$ of layer L_{i+1} that are enclosed by the boundary cycle $B(C_{i+1,j})$. For each pair $\tilde{y}, \tilde{z} \in B(C_{i+1,j}) \cap N(x)$ (where $\tilde{y} \neq \tilde{z}$), we consider the path $P_{\tilde{y},\tilde{z}}$ from $\tilde{y}$ to $\tilde{z}$ along the cycle $B(C_{i+1,j})$, taking the direction such that the region enclosed by $\{\tilde{z}, x\}$, $\{x, \tilde{y}\}$, and $P_{\tilde{y},\tilde{z}}$ contains the layer-component $C_{i+1,j'}$. Let $\{y, z\} \subseteq B(C_{i+1,j}) \cap N(x)$ be the pair such that the corresponding path $P_{y,z}$ is shortest. The triple, then, is the three-element set $\{x, y, z\}$. If x has no or only a single neighbor y in $B(C_{i+1,j})$, then the "triple" consists only of $\{x\}$, or $\{x, y\}$.
For each vertex $x \in C_{i+1,j}$ of D_{i+1} and each non-empty layer-component $C_{i+1,j'}$ that is enclosed by $B(C_{i+1,j})$, we obtain such a lower triple.

Definition 8 A *middle triple* for layer L_i is associated to a non-empty layer-component $C_{i+1,j}$ and a vertex $x \in D_i$ that has a neighbor in $B(C_{i+1,j})$ (see Fig. 3). Note that, due to the layer model, it is easy to see that a vertex $x \in D_i$ can have at most two neighbors y, z in $B(C_{i+1,j})$. Depending on whether x itself lies on the cycle $B(C_{i+1,j})$ or not, we obtain two different cases which both are illustrated in Fig. 3. In either of these cases the middle triple is defined as the set $\{x, y, z\}$. Again, if x has none or only a single neighbor y in $B(C_{i+1,j})$, then the "triple" consists only of $\{x\}$, or $\{x, y\}$.
For each non-empty layer-component $C_{i+1,j}$ and each vertex $x \in D_i$, we obtain such a middle triple.

Definition 9 We define the set S_i as the union of all upper triples, lower triples and middle triples of L_i.

In the following, we will show that S_i is a separator of the graph. Note that the upper bounds on the size of S_i, which are derived afterwards, are crucial for the upper bound on the treewidth derived later on.

Theorem 10. *The set S_i separates vertices of L_{i-1} and L_{i+2}.*

Proof. Suppose there is a path P (with no repeated vertices) from layer L_{i+2} to layer L_{i-1} that avoids S_i. This clearly implies that there exists a path P' from a vertex x in a non-empty layer-component $C_{i+1,j}$ of layer L_{i+1} to a vertex $z \in B(B(C_{i+1,j}))$ in layer L_{i-1} which has the following two properties:

- $P' \cap S_i = \emptyset$.
- All vertices in between x and z belong to layer L_i or to empty layer-components of layer L_{i+1}.

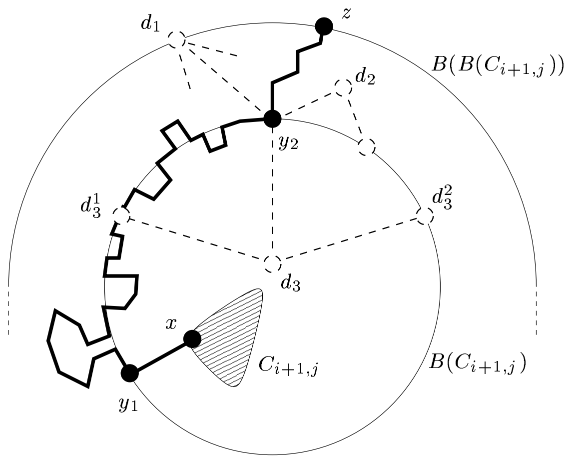


Fig. 4. S_i separates L_{i-1} and L_{i+2}.

(This can be achieved by simply taking a suitable subpath P' of P.) Let y_1 (and y_2, respectively) be the first (last) vertex along the path P' from x to z that lies on the boundary cycle $B(C_{i+1,j}) \subseteq L_i$ (see Fig. 4).

Obviously, y_2 cannot be an element of D, since, then, it would appear in a middle triple of layer L_i and, hence, in S_i. We now consider the vertex that dominates y_2. This vertex can lie in layer L_{i-1}, L_i or L_{i+1}.

Suppose first that y_2 is dominated by a vertex $d_1 \in L_{i-1}$. Then d_1 is in $B(B(C_{i+1,j}))$, simply by definition of the boundary cycle (see Fig. 4). Since G is planar, this implies that y_2 must be an "outermost" neighbor of d_1 among all elements in $N(d_1) \cap B(C_{i+1,j})$. If this were not the case, then there would be an edge from d_1 to a vertex on $B(C_{i+1,j})$ that leaves the closed region bounded by $\{d_1, y_2\}$, the path from y_2 to z, and the corresponding path from z to d_1 along $B(B(C_{i+1,j}))$. Hence, y_2 is in the upper triple of layer L_i which is associated to the layer-component $C_{i+1,j}$ and d_1. This contradicts the fact that P' avoids S_i.

Now, suppose that y_2 is dominated by a vertex $d_2 \in D_i$ (see Fig. 4). By definition of the middle triples, this clearly implies that y_2 is in the middle triple associated to $C_{i+1,j}$ and d_2. Again, this contradicts the assumption that $P' \cap S_i = \emptyset$.

Consequently, the dominating vertex d_3 of y_2 has to lie in layer L_{i+1}. Let $\{d_3, d_3^1, d_3^2\}$, where $d_3^1, d_3^2 \in N(d_3) \cap B(C_{i+1,j})$, be the lower triple associated to layer-component $C_{i+1,j}$ and d_3 (see Fig. 4). By definition, $C_{i+1,j}$ is contained in the region enclosed by $\{d_3^1, d_3\}, \{d_3, d_3^2\}$ and the path from d_3^2 to d_3^1 along $B(C_{i+1,j})$, which—assuming that $y_2 \notin \{d_3, d_3^1, d_3^2\}$—does not hit y_2 (see Fig. 4). We now observe that, whenever the path from y_1 to y_2 leaves the cycle $B(C_{i+1,j})$ to its exterior, say at a vertex q, then it has to return to $B(C_{i+1,j})$ at a vertex $q' \in N(q) \cap B(C_{i+1,j})$. This, however, shows that the path P' has to hit either

d_3^1 or d_3^2 on its way from y_1 to y_2. Since $d_3^1, d_3^2 \in S_i$, this case also contradicts the fact that $P' \cap S_i = \emptyset$. □

Lemma 1. $|S_i| \leq 5(t_{i-1} + t_i + t_{i+1}) + 12c_{i+1}$.

Proof. We give bounds for the number of vertices in upper, middle and lower triples of layer i, separately.

Firstly, we discuss the upper triples of layer i, which were associated to a non-empty layer-component $C_{i+1,j}$ of layer L_{i+1} and a vertex $x \in D_{i-1}$ with neighbors in $B(C_{i+1,j})$. Consider the bipartite graph G' which has vertices for each non-empty layer-component $C_{i+1,j}$ and for each vertex in D_{i-1}. Whenever a vertex in D_{i-1} has a neighbor in $B(C_{i+1,j})$, an edge is drawn between the corresponding vertices in G'. Each edge in G', by construction, may correspond to an upper triple of layer L_i. Note that G' is a planar bipartite graph whose bipartition subsets consist of t_{i-1} and c_{i+1} vertices, respectively. Thus, the number of edges of G' is linear in the number of vertices; more precisely, it is bounded by $2(t_{i-1} + c_{i+1})$. From this, we obtain an upper bound for the number of vertices in upper triples of layer L_i as follows: Potentially, each vertex of D_{i-1} appears in an upper triple and, for each edge in G', we possibly obtain two further vertices in an upper triple. This shows that the total number of vertices in upper triples is bounded by $t_{i-1} + 4(t_{i-1} + c_{i+1})$.

A similar analysis can be used to show that the number of vertices in the lower triples is bounded by $t_{i+1} + 4(t_{i+1} + c_{i+1})$ and that the number of vertices in the middle triples can be bounded by $t_i + 4(t_i + c_{i+1})$.

By definition of S_i, this proves our claim. □

Note that, by a more detailed investigation, the bound given in Lemma 1 probably can be improved. One observes, e.g., that the planar bipartite graph G', which was constructed in the proof, has the special property that it is a "hyperplane" bipartite graph, i.e., one of the bipartition subsets can be arranged on a line and all edges of the graph lie in one halfplane of this line. This property of G' is immediate from the fact that the upper triples associated to a non-empty layer-component $C_{i+1,j}$ lie within the boundary cycle $B(B(C_{i+1,j}))$. For such graphs, first investigations indicate that one can obtain better estimates on the number of their edges than the ones used in the proof of Lemma 1.

A similar observation can be made for estimating the bounds for the lower triples.

Lemma 2. $c_i \leq t_i + t_{i+1} + t_{i+2}$.

Proof. By definition, c_i refers to only non-empty layer-components in layer L_i, i.e., there is at least one vertex of layer L_{i+1} contained within each such layer-component. Such a vertex can only be dominated by a vertex from layer L_i, L_{i+1}, or L_{i+2}. In this way, we get the claimed upper bound. □

Lemma 3. $\sum_{i=1}^{r} |S_i| \leq 51k$, *where r is the number of layers of the graph.*

Proof. This follows directly when we combine the previous two lemmas. □

Consider the following three sets of vertices: $\mathbb{S}_0 = S_1 \cup S_4 \cup S_7 \cup \ldots$, $\mathbb{S}_1 = S_2 \cup S_5 \cup S_8 \cup \ldots$ and $\mathbb{S}_2 = S_3 \cup S_6 \cup S_9 \cup \ldots$. As $|\mathcal{S}_1| + |\mathcal{S}_2| + |\mathcal{S}_3| \leq 51k$, one of these sets has size at most $\frac{51}{3}k$, say $\mathcal{S}_\delta$ (with $\delta \in \{0, 1, 2\}$).

Theorem 11. *A planar graph with domination number k has treewidth of at most $6\sqrt{34}\sqrt{k}$.*

Proof. Let δ and $\mathbb{S}_\delta$ be as obtained above. Let $d := \frac{3}{2}\sqrt{34}$. We now go through the sequence $S_{1+\delta}, S_{4+\delta}, S_{7+\delta}, \ldots$ and look for separators of size at most $s(k) := d\sqrt{k}$. Due to the estimate on the size of $\mathbb{S}_\delta$, such separators of size at most $s(k)$ must appear within each $n(k) := \frac{51}{3}d^{-1}\sqrt{k} = \frac{1}{3}\sqrt{34}\sqrt{k}$ sets in the sequence. In this manner, we obtain a set of disjoint separators of size at most $s(k)$ each, such that any two consecutive separators from this set are at most $3n(k)$ layers apart. Clearly, the separators chosen in this way fulfil the requirements in Proposition 4.

Observe that the components cut out in this way each have at most $3n(k)$ layers and, hence, their treewidth is bounded by $9n(k)$ due to Proposition 2.

Using Proposition 4, we can compute an upper bound of the treewidth tw of the originally given graph with domination number k:

$$\begin{aligned} \mathrm{tw}(k) &\leq 2s(k) + 9n(k) \\ &= 2(\frac{3}{2}\sqrt{34}\sqrt{k}) + 9(\frac{1}{3}\sqrt{34}\sqrt{k}) \\ &= 6\sqrt{34}\sqrt{k}. \end{aligned}$$

This proves our claim. □

Observe that the tree structure of the tree decomposition obtained in the preceding proof corresponds to the structure of the layer decomposition forest.

How did we come to the constants? We simply computed the minimum of $2s(k) + 9n(k)$ (the upper bound on the treewidth) given the bound $s(k)n(k) \leq \frac{51}{3}k$. This suggests $s(k) = d\sqrt{k}$, and d is optimal when $2s(k) = 9n(k) = 9 \cdot \frac{51}{3} \cdot k \cdot s(k)^{-1}$, so, $2d = \frac{153}{d}$, i.e., $d = \frac{3}{2}\sqrt{34}$.

As already mentioned above, it seems to be possible to improve upon the bound of the treewidth by a more refined analysis.

3.2 Tree decomposition

The proofs above can be turned into constructive algorithms that find tree decompositions of the stated widths. From the proof in [5] that an r-outerplanar graph has treewidth at most $3r - 1$, one can construct a linear time algorithm that indeed finds a tree decomposition of width $3r - 1$ of a given r-outerplanar graph. The proofs in this paper can also be made constructive, but there is one point that needs specific attention. As we do not start with the dominating set given, we cannot construct the upper, middle, and lower triples. Instead, we compute the minimum separator between L_{i-1} and L_{i+2} directly, and use that set instead of S_i as defined in the proof of Section 3.1. Such a minimum separator can be computed with well known techniques based on maximum flow (see e.g., [20]). The running time to find one such separator is $O(sn')$, where s is the size of the separator, and n' the number of vertices that are involved. The total time to find all separators, stopping when separators become so large that they will not be used further in the algorithm, can be bounded by $O(\sqrt{k}n)$.

Theorem 12. *The* PLANAR DOMINATING SET *problem can be solved in time* $O(c^{\sqrt{k}}n)$, *where* k *is the domination number of the given graph of size* n, *and* $c = 3^{6\sqrt{34}}$.

Proof. A tree decomposition of width $6\sqrt{34}\sqrt{k}$ of G can be constructed in $O(\sqrt{k}n)$ time. (If k is not known in advance, then an $O(\sqrt{k}n)$ time algorithm is still possible for this step, using detailed bookkeeping techniques. Otherwise, one can try different values of k – this can be done at the cost of an extra multiplicative factor of $O(\log k)$ by using binary search.) Then, this tree decomposition can be used to solve the DOMINATING SET problem, as described in Theorem 4. □

The constant c above is $3^{6\sqrt{34}}$, which is rather large. However, a more refined analysis will help to reduce this constant significantly. Moreover, it is a worst case estimate, which might be far from what happens in practical applications.

4 Variations of DOMINATING SET and DISK DIMENSION

For several variations of the DOMINATING SET problem, our technique can also help to obtain algorithms with a similar running time. In particular, we have the following. Let DOMINATING SET WITH PROPERTY P be the following graph problem: Given a graph $G = (V, E)$, find the minimum size set $W \subseteq V$ with W a dominating set and where property $P(W)$ holds.

Theorem 13. *Suppose there is an algorithm that solves in* $O(q^\ell \cdot n)$ *time the* DOMINATING SET WITH PROPERTY P *problem on graphs, given a tree decomposition with treewidth* ℓ *and* n *nodes for some constant* q. *Then the* DOMINATING SET WITH PROPERTY P *problem can be solved in* $O(q^{d\sqrt{k}} \cdot n)$ *time on planar graphs, where* k *is the minimum size dominating set with property* P *and* $d = 6\sqrt{34}$.

Proof. If the planar graph G admits a dominating set with property P of size at most k, then, clearly, G has domination number at most k. By Theorem 11, the treewidth of G is bounded by $6\sqrt{34}\sqrt{k}$. According to the discussion in Section 3.2, a corresponding tree decomposition can be found in time $O(\sqrt{k}n)$. The assumption on the existence of an $O(q^\ell \cdot n)$ time algorithm for given tree decomposition of width ℓ then yields the claim. □

Problems for which the condition of Theorem 13 holds and, hence, for which we can find such an $O(c^{\sqrt{k}} \cdot n)$ time algorithm are, for instance, the INDEPENDENT DOMINATING SET problem, TOTAL DOMINATING SET problem, or CONNECTED DOMINATING SET problem.

We now turn our attention to the DISK DIMENSION problem (see [2, 14, 25]) which is the following: Given a plane graph $G = (V, E)$ (i.e., a planar graph with a fixed embedding), find the minimum set of faces that cover all vertices of G. We can use the techniques established for solving DOMINATING SET WITH PROPERTY P on planar graphs to solve the DISK DIMENSION problem:

Let $G = (V, E)$ be a plane graph. Consider the following graph: Add a vertex to each face of G, and make each such "face vertex" adjacent to all vertices that are on the boundary of that face. Let $G' = (V', E')$ be the resulting graph. Write $V' = V \cup V_F$, where V_F is the set of vertices that represent a face in G.

For $W \subseteq V'$, we define $P'(W) = \texttt{true}$ if and only if $W \subseteq V_F$. Then, by construction, there is a one-to-one correspondence between the sets of faces that cover the vertices of G and dominating sets in G' with property P'. In this sense, the DISK DIMENSION problem can be transformed to the DOMINATING SET WITH PROPERTY P' problem in linear time.

Theorem 14. *The* DISK DIMENSION *problem can be solved in time $O(c_1^{\sqrt{k}} n)$, where k is the disk dimension of the given graph of size n, and $c_1 = 2^{6\sqrt{34}}$.*

Proof. Consider the graph $G' = (V', E')$ with $V' = V \cup V_F$ as given above. Given a tree decomposition of width ℓ, the dominating set problem with property P' can be solved in time $O(2^{\ell} \cdot n)$, similar to the dynamic programming algorithm sketched in the proof of Theorem 4. Observe that the size of the tables we have to use for each bag are smaller than for the general dominating set problem, since each vertex of V_F is either in the dominating set or not and each vertex of V is either dominated or not. This gives table size 2^{ℓ}. Theorem 13 and the one-to-one correspondence between this problem and the disk dimension problem yield the claim. □

We remark that the problem DOMINATING SET WITH PROPERTY P' as defined above is, in a bipartite variant, bascially called PLANAR RED/BLUE DOMINATING SET in [14, p.38]. There, Downey and Fellows derive an $O(12^k n)$ algorithm for this problem. In the same place, they give an $O(12^k n)$ algorithm for DISK DIMENSION, which they call FACE COVER NUMBER FOR PLANAR GRAPHS. Hence, our observations lead to asymptotic improvements of their results.

5 Conclusion

In this paper, we presented a treewidth-based approach to improve the fixed parameter complexity of the PLANAR DOMINATING SET and the DISK DIMENSION problem drastically—we gained an exponential improvement over previous exact solutions for the problems. Seemingly for the first time, our results provide fixed parameter algorithms whose exponential factor has an exponent sublinear in the parameter.

In the long version of this paper, we plan to give improved estimates for the constant bases of the exponential terms. In addition, it would be interesting to investigate the practical usefulness of our result, since our estimates for the constants are worst case and very pessimistic ones. It also is interesting to see if these results can be extended to more variants of Dominating Set and to other graph classes (e.g., graphs of bounded genus). Another interesting open problem is how to use the techniques of this paper for the variant of the disk dimension problem, where the embedding is not given as ab input (i.e., for a given planar graph, find an embedding with minimum number of faces that cover all the vertices).

Finally, we remark that similar results on planar dominating set and related problems can be obtained by making use of the small separator techniques presented in this paper together with the algorithms for outerplanarity-bounded graphs developed by Baker [1], which would also yield running times of the form $O(c^{\sqrt{k}}n)$ for some constant c, where k is the domination number of the given graph.

Acknowledgements

We thank Ton Kloks for many discussions on the topic of this paper.

References

1. B. S. Baker, Approximation algorithms for NP-complete problems on planar graphs, *Journal of the ACM* **41**:153–180, 1994.
2. D. Bienstock and C. L. Monma, On the complexity of covering vertices by faces in a planar graph. *SIAM J. Comput.* **17**(1):53–76, 1988.
3. H. L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* **25**:1305–1317, 1996.
4. H. L. Bodlaender, Treewidth: Algorithmic techniques and results. In *Proceedings 22nd MFCS'97*, Springer-Verlag LNCS 1295, pp. 19–36, 1997.
5. H. L. Bodlaender, A partial k-arboretum of graphs with bounded treewidth. *Theor. Comp. Sci.* **209**:1–45, 1998.
6. A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes: a Survey*, SIAM Monographs on Discrete Mathematics and Applications, 1999.
7. H. J. Broersma, T. Kloks, D. Kratsch, and H. Müller, Independent sets in AT-free graphs, In *Proceedings ICALP'97*, Springer-Verlag LNCS 1256, pp. 760–770, 1997.
8. M. S. Chang, Efficient algorithms for the domination problem on interval and circular arc graphs, *SIAM J. Comput.* **27**:1671–1694, 1998.
9. J. Chen, I. Kanj, and W. Jia, Vertex cover: further observations and further improvements. In *Proceedings 25th WG*, Springer-Verlag LNCS 1665, pp. 313–324, 1999.
10. D. G. Corneil and L. K. Stewart, Dominating sets in perfect graphs, *Discrete Mathematics* **86**, (1990), 145–164.
11. P. Crescenzi and V. Kann, A compendium of NP optimization problems. Available at http://www.nada.kth.se/theory/problemlist.html, August 1998.
12. M. Damian-Iordache and S.V. Pemmaraju, A $(2+\epsilon)$-approximation scheme for minimum domination on circle graphs. In *Proceedings 11th ACM-SIAM SODA 2000*, pp. 672–679.
13. R. G. Downey and M. R. Fellows, Parameterized computational feasibility. In *P. Clote, J. Remmel (eds.): Feasible Mathematics II*, pp. 219–244. Birkhäuser, 1995.

14. R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Springer-Verlag, 1999.
15. R. D. Dutton and R. C. Brigham, Domination in claw-free graphs, *Congr. Num.* **132**: 69–75, 1998.
16. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman, 1979.
17. T. W. Haynes, S. T. Hedetniemi, and P. J. Slater (eds.), *Domination in graphs*, Marcel Dekker, 1998.
18. M. A. Henning, Domination in graphs, A survey, *Congr. Num.* **116**:139–179, 1996.
19. D. S. Hochbaum (ed.), *Approximation algorithms for NP-hard problems.* PWS Publishing Company, 1997.
20. A. Kanevsky, Finding all minimum-size separating vertex sets in a graph, *Networks* **23**: 533–541, 1993.
21. J. van Leeuwen, Graph Algorithms, In *Handbook of Theoretical Computer Science, Vol. A: Algorithms and Complexity Theory*, North Holland, 1990.
22. H. Müller and A. Brandstädt, The NP-completeness of Steiner tree and dominating set for chordal bipartite graphs, *Theor. Comp. Sci.* **53**:257–265, 1987.
23. R. Niedermeier and P. Rossmanith. Upper bounds for Vertex Cover further improved. In *Proceedings 16th STACS*, Springer-Verlag LNCS 1563, pp. 561–570, 1999.
24. C. H. Papadimitriou, *Computational Complexity.* Addison-Wesley, 1994.
25. R. C. Read, Prospects for graph theory algorithms. *Ann. Discr. Math.* **55**:201–210, 1993.
26. J. A. Telle, Complexity of domination-type problems in graphs, *Nordic J. Comput.* **1**:157–171, 1994.
27. J. A. Telle and A. Proskurowski, Algorithms for vertex partitioning problems on partial k-trees, *SIAM J. Discr. Math.* **10(4)**:529–550, 1997.