Bidimensionality: New Connections between FPT Algorithms and PTASs

Erik D. Demaine* MohammadTaghi Hajiaghayi*

Abstract

We demonstrate a new connection between fixed-parameter tractability and approximation algorithms for combinatorial optimization problems on planar graphs and their generalizations. Specifically, we extend the theory of so-called "bidimensional" problems to show that essentially all such problems have both subexponential fixed-parameter algorithms and PTASs. Bidimensional problems include e.g. feedback vertex set, vertex cover, minimum maximal matching, face cover, a series of vertex-removal problems, dominating set, edge dominating set, r-dominating set, diameter, connected dominating set, connected edge dominating set, and connected r-dominating set. We obtain PTASs for all of these problems in planar graphs and certain generalizations; of particular interest are our results for the two well-known problems of connected dominating set and general feedback vertex set for planar graphs and their generalizations, for which PTASs were not known to exist. Our techniques generalize and in some sense unify the two main previous approaches for designing PTASs in planar graphs, namely, the Lipton-Tarjan separator approach [FOCS'77] and the Baker layerwise decomposition approach [FOCS'83]. In particular, we replace the notion of separators with a more powerful tool from the bidimensionality theory, enabling the first approach to apply to a much broader class of minimization problems than previously possible; and through the use of a structural backbone and thickening of layers we demonstrate how the second approach can be applied to problems with a "nonlocal" structure.

1 Introduction

The recent theory of fixed-parameter algorithms and parameterized complexity [32] has attracted much attention in its less than ten years of existence. In general the goal is to understand when the exponentiality of NP-hard problems can be contained within a parameter of the problem that in some cases is independent of the problem size. Fixed-parameter algorithms whose running time is polynomial for fixed parameter values make these problems efficiently solvable whenever the parameter is reasonably small. In several applications, e.g., finding locations to place fire stations, we prefer exact solutions at the cost of running time: we can afford high running time (e.g., several weeks of real time) if the resulting solution builds fewer fire stations (which are extremely expensive).

A general result of Cai and Chen [16] says that if an NP optimization problem has an FPTAS, i.e., a PTAS with running time $(1/\varepsilon)^{O(1)}n^{O(1)}$, then it is fixed-parameter tractable. Others [10, 17] have generalized this result to any problem with an EPTAS, i.e., a PTAS with running time $f(1/\varepsilon)n^{O(1)}$ for any function f. On the other hand, no reverse transformation is possible in general, because for example vertex cover is an NP optimization problem that is fixed-parameter tractable but has no PTAS in general graphs (unless $P = NP$).

Nonetheless, in this paper, we present a general (reverse) transformation from fixed-parameter algorithms to PTASs for a broad class of optimization problems in planar graphs and their generalizations.

In the last three years, several researchers have obtained exponential speedups in fixed-parameter algorithms for various problems on several classes of graphs. While most previous fixed-parameter algorithms have a running time of $O(2^{O(k)}n^{O(1)})$ or worse, the exponential speedups result in subexponential algorithms with typical running times of $O(2^{O(\sqrt{k})}n^{O(1)})$. For example, the first fixed-parameter algorithm for finding a dominating set of size k in planar graphs [2] has running time $O(8^k n)$; subsequently, a sequence of subexponential algorithms and improvements have been obtained, starting with running time $O(4^{6\sqrt{34k}}n)$ [1], then $O(2^{27\sqrt{k}}n)$ [47], and finally $O(2^{15.13\sqrt{k}}k + n^3 + k^4)$ [36]. Other subexponential algorithms for other domination and covering problems on planar graphs have also been obtained [1, 3, 18, 50, 46].

However, all of these algorithms apply only to planar graphs. In another sequence of papers, these results have been generalized to other classes of graphs that include planar graphs: map graphs [22], bounded-genus graphs [24], single-crossing-minor-free graphs [29, 30], apex-minor-free graphs [23, 26], and H-minor-free graphs [24]. These algorithms [22, 24, 30, 29, 23, 26] apply to several combinatorial

*MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, Massachusetts 02139, USA, {edemaine, hajiagha}@mit.edu.

optimization problems related to domination and covering.

All subexponential fixed-parameter algorithms developed so far are based on showing a "treewidth-parameter bound": any graph with parameter value k has treewidth at most some function $f(k)$. (A *parameter* simply assigns a nonnegative integer to every graph.) In many cases, $f(k)$ is sublinear in k, often $O(\sqrt{k})$. Combined with algorithms that are singly exponential in treewidth and polynomial in problem size, such a bound immediately leads to subexponential fixed-parameter algorithms.

Essentially all treewidth-parameter bounds proved so far are captured by the broad class of "bidimensional" problems introduced in a series of papers [30, 22, 24, 23]. Roughly speaking, a parameterized problem is *bidimensional* if the parameter is large (e.g., linear) in a grid and closed under contractions (*contraction-bidimensional*) or closed under minors (*minor-bidimensional*). (A parameter is *closed* under an operation if performing that operation on a graph never increases the parameter value.) Examples of bidimensional problems include e.g. feedback vertex set, vertex cover, minimum maximal matching, face cover, a series of vertex-removal problems, dominating set, edge dominating set, r-dominating set, diameter, connected dominating set, connected edge dominating set, connected r-dominating set, and planar set cover. Treewidth-parameter bounds have been established for all bidimensional problems in planar graphs [22], bounded-genus graphs [24], single-crossing-minor-free graphs [30], and apex-minor-free graphs [23, 25], and for all minor-bidimensional problems in H-minor-free graphs [24]. In particular, the established bound is sublinear for planar graphs, bounded-genus graphs, single-crossing-minor-free graphs, and in some cases for apex-minor-free graphs. In summary, bidimensionality is the most powerful method so far for establishing treewidth-parameter bounds and therefore for designing subexponential fixed-parameter algorithms, encompassing all such previous results for planar graphs and their generalizations.

In this paper, we demonstrate that bidimensionality allows us to not only design fast fixed-parameter algorithms but also to design fast PTASs. More precisely, we prove that any bidimensional problem satisfying a few straightforward constraints not only has a subexponential fixed-parameter algorithm but also has a PTAS for planar graphs and some generalizations. Thus bidimensionality enables us to easily obtain both subexponential fixed-parameter algorithms and PTASs for a wide variety of problems in planar graphs and their generalizations, and provides a connection between fixed-parameter tractability and approximation in this setting. In particular, our results lead to new PTASs for several well-known problems that were previously not known to have PTASs on planar graphs. Our novel approach of using tools from fixed-parameter tractability to design PTASs can be considered as the reverse of the layerwise-separation approach in [1, 34, 37] which uses tools from approximation (Baker's approach) to design fixed-parameter algorithms.

Our original motivation was that the bidimensionality theory almost trivially gave us subexponential fixed-parameter algorithms for some minor-bidimensional problems, such as general feedback vertex set, yet PTASs for these problems in planar graphs and their generalizations such as single-crossing-minor-free graphs remained elusive. Here we obtain an EPTAS for general feedback vertex set in planar graphs and more generally single-crossing-minor-free graphs as a simple by-product of our general approach for minor-bidimensional parameters. Another motivating problem is connected dominating set, which is bidimensional, yet lacks a fast enough bounded-treewidth algorithm for the theory to apply; indeed, the existence of subexponential fixed-parameter algorithms for connected dominating set in planar graphs was implicitly asked by Alber et al. [1]. Here we not only establish a subexponential fixed-parameter algorithm for this problem (see Theorem 8.1) but also use our machinery to obtain a PTAS for the same problem, which was not previously known to exist.

While our focus is on our general techniques, we point out that the two problems mentioned above—general feedback vertex set and connected dominating set—are important problems that have been studied extensively in the literature. Feedback vertex set—finding a minimum-size set F of vertices whose removal leaves the graph acyclic—is a basic problem in graph algorithms with applications to e.g. deadlock resolution. The first approximation algorithms for this problem were a $O(\lg n)$-approximation for general graphs and a 10-approximation for planar graphs [9]. Subsequently, 2-approximation algorithms for general graphs have been discovered [7, 11]. Goemans and Williamson [39] apply the primal-dual method to obtain a $(9/4)$-approximation for this problem. Although $9/4 > 2$, the LP relaxation they consider has interesting implications on the Akiyama-Watanabe Conjecture about the size of a feedback vertex set in a planar graph. Their results also apply to a generalized form of feedback vertex set. The approximation factor of the primal-dual method in undirected planar graphs has been further improved to two (see e.g., [21]). Connected dominating set—finding a minimum-size set D of vertices such that every vertex not in D is adjacent to at least one vertex in D and in addition the subgraph induced by D is connected—is a fundamental problem in connected facility location, a basic problem in operations research and computer science; see e.g. [57, 48, 45]. Another more recent application of this problem is in finding a "virtual backbone-based routing strategy" in a wireless ad-hoc network; see e.g. [4, 58]. The first and so-far best approximation algorithm in general graphs is the $(\ln \Delta + O(1))$-approximation of Guha and Khuller [44], where Δ is the maximum degree in the graph. For unit-disk graphs, several approximation algorithms have been developed (see e.g. [4, 52]), culminating with a recent PTAS [19].

There are two main general approaches for designing PTASs for problems on planar graphs. The first approach is based on planar separators [51]. The approximation algorithms resulting from this approach are generally impractical; for example, just to achieve an approximation ratio of 2, the base case of the planar-separator approach requires exhaustive solution of graphs of up to $2^{2^{400}}$ vertices [20]. To address this impracticality, Baker [8] introduced the second approach for PTASs in planar graphs, based on decomposition into overlapping subgraphs of bounded outerplanarity. Specifically, Baker's approach obtains $(1+\varepsilon)$-approximation algorithms with running times of $2^{O(1/\varepsilon)}n^{O(1)}$ for many problems on planar graphs, such as maximum independent set, minimum dominating set, and minimum vertex cover. Eppstein [34, 33] generalized Baker's approach to a broader class of graphs called graphs of bounded local treewidth, i.e., where the treewidth of the subgraph induced by the set of vertices at distance at most r from any vertex is bounded above by some function $f(r)$ independent of n. Recently there has been much work on graphs of bounded local treewidth [37, 43, 29, 26, 24, 34, 23]. In particular, Eppstein [34] characterized all minor-closed families of graphs that have bounded local treewidth, showing that they are precisely apex-minor-free graphs, where an *apex graph* has a vertex whose removal leaves a planar graph. Khanna and Motwani [49] use Baker's approach in an attempt to syntactically characterize the complexity class of problems admitting PTASs, establishing a family of problems on planar graphs to which it applies. Frick and Grohe [37] use Baker's approach to obtain efficient (near-linear) algorithms to decide arbitrary properties definable in first-order logic.

Unfortunately, both of these approaches for PTASs in planar graphs seem to be limited, at least in their current use. In the separation approach, the separator is bounded in terms of n ($O(\sqrt{n})$), which can be large compared to the cost of the optimal solution. As a result, the approach has been used so far only in a few limited minimization problems (to the best of our knowledge, just vertex cover [53] and forms of TSP [41, 6, 40, 42]) where, after some graph reductions (linear kernelization), the cost of the optimum solution can be lower bounded in terms of n. For example, Grohe [43] states that dominating set is a problem "to which the technique based on the separator theorem does not apply". On the other hand, all applications of Baker's approach so far are to optimization problems arising from "local" properties (such as those definable in first-order logic). Intuitively, such local properties can be decided by locally checking every constant-size neighborhood. In particular, this restriction has limited attempts at characterizing the complexity class of problems admitting PTASs [37, 49].

In this paper we demonstrate that the bidimensionality theory enables us to bypass these limitations and generalize both approaches.

First in Sections 4–5 we generalize the separation approach to obtain PTASs for all bidimensional problems that satisfy a few straightforward constraints, and to generalizations of planar graphs. In particular, this includes all problems and graph classes for which subexponential fixed-parameter algorithms have been obtained. Our technique is based on evenly dividing the optimum solution instead of the whole graph, using a tree decomposition found by treewidth-approximation algorithms for certain classes of graphs, and using the small treewidth guaranteed by bidimensionality. Evenly dividing the optimum solution is difficult because we do not know the optimum solution; nonetheless, we show that such a division can be done approximately using existing constant-factor (or even logarithmic-factor) approximations. We also use the fast fixed-parameter algorithms from the bidimensionality theory to remove an extra log factor in the exponent of the running time. Through our approach we immediately obtain an EPTAS for general feedback vertex set in planar and more generally single-crossing-minor-free graphs. Combined with our fixed-parameter results mentioned above, we obtain a PTAS for connected dominating set in planar and single-crossing-minor-free graphs. For these problems in bounded-genus graphs and apex-minor-free graphs, we also obtain "almost PTASs" with almost-polynomial running time $n^{O(\lg \lg n)}$ for fixed ε.[1] We refer the reader to Corollaries 4.3 and 5.1 for a complete list of important problems for which we obtain new PTASs and almost PTASs.

Second in Section 6 we generalize Baker's approach (which is generally considered faster than the previous approach) to obtain PTASs for nonlocal problems using two main techniques. Our first technique is to use a constant-factor (or even logarithmic-factor) approximation to the problem as a "backbone" for achieving the needed nonlocal property. Of course, we cannot use the entire approximate solution, so we take a $\Theta(\varepsilon)$ fraction by slicing at (intersecting with) a small number of layers in the graph and removing the rest. Now we are left with two challenges: we need to restore the nonlocal property of the full backbone, and we need the subproblems in the layers between these slices to form a global solution comparable to the overall optimum. The second technique addresses both of these problems by using thicker subproblems extending beyond the slices by $\Theta(\log n)$ layers instead of the usual $\Theta(1)$ in Baker's approach. Of course, the devil is in the details. We are left with the task of solving the subproblems, which are a generalized form of the original problem in order to restore the nonlocal property of the backbone. For connected dominating set, these generalized subproblems can be solved using our fixed-

[1]This time bound is substantially better than the existing notion of quasipolynomial time, $n^{O(\lg n)}$. Also, $\lg \lg n$ is at most 8 for $n \leq 2^{256}$, which is nearly the number of particles in the known universe.

parameter algorithm mentioned above. A final challenge is that the running time of this algorithm is superpolynomial, $(\lg n)^{\Theta(\lg n)}$, because the thickness of a subproblem is now a function of n, $\Theta(\lg n)$, instead of a constant as in Baker's approach. Using planarity of the subproblems, or more specifically their low outerplanarity, together with properties of a simple, direct, and efficient construction of a tree decomposition for such graphs of low outerplanarity [1, 13], we obtain in Section 7 more efficient encodings of subproblems and reduce the running time to $2^{\Theta(\lg n)} = n^{O(1)}$ for fixed ε.

Last but not least, bidimensionality provides new strong connections between fixed-parameter algorithms, approximation algorithms, and the two existing approaches to finding PTASs in planar graphs and their generalizations. In particular, essentially every bidimensional problem has both a subexponential fixed-parameter algorithm and a PTAS in such graphs. At a deeper level, Baker's approach itself can be viewed as a special case of the bidimensional theory, as it is just a combination of a "shifting strategy" and the bidimensionality of the diameter of a graph [25]. The bidimensionality of diameter, or more precisely the resulting parameter-treewidth relation, was the point of Eppstein's work [34].

2 Definitions and Preliminary Results

Our graph terminology is as follows. All graphs are finite, simple, and undirected, unless indicated otherwise. For a graph G, we denote its vertex set by $V(G)$ and its edge set by $E(G)$. Let $n = |V(G)|$ denote the number of vertices when G is clear from context. For every nonempty $W \subseteq V(G)$, the subgraph of G induced by W is denoted by $G[W]$. We define the *r-neighborhood* of a vertex set $S \subseteq V(G)$, denoted by $N_G^r(S)$, to be the set of vertices at distance at most r from at least one vertex of $S \subseteq V(G)$; if $S = \{v\}$ we simply use the notation $N_G^r(v)$. The *diameter* of G, denoted by $diam(G)$, is the maximum over all distances between pairs of vertices of G. We assume the reader is familiar with other general concepts of graph theory such as directed graphs, trees, and planar graphs. The reader is referred to standard references for appropriate background [14]. In addition, for exact definitions of various NP-hard graph-theoretic problems in this paper, the reader is referred to Garey and Johnson [38].

Given an edge $e = \{x, y\}$ of a graph G, the graph G/e is obtained from G by contracting the edge e; that is, to get G/e we identify the vertices x and y and remove all loops and duplicate edges. A graph H obtained by a sequence of edge contractions is said to be a *contraction* of G. A graph H is a *minor* of a graph G, denoted $H \preceq G$, if H is a subgraph of a contraction of G. A graph class $\mathcal{C}$ is *minor-closed* if any minor of any graph in $\mathcal{C}$ is also a member of $\mathcal{C}$. A minor-closed graph class $\mathcal{C}$ is *H-minor-free* if $H \notin \mathcal{C}$. For example, a planar graph is a graph excluding both $K_{3,3}$ and K_5 as minors.

The notion of treewidth was introduced by Robertson and Seymour [54] and plays an important role in their fundamental work on graph minors. To define this notion, first we consider the representation of a graph as a tree, which is the basis of our algorithms.

DEFINITION 2.1. ([54]) *A* tree decomposition *of a graph $G = (V, E)$, denoted by $TD(G)$, is a pair (χ, T) in which $T = (I, F)$ is a tree and $\chi = \{\chi_i \mid i \in I\}$ is a family of subsets of $V(G)$ such that: (1) $\bigcup_{i \in I} \chi_i = V$; (2) for each edge $e = \{u, v\} \in E$ there exists an $i \in I$ such that both u and v belong to χ_i; and (3) for all $v \in V$, the set of nodes $\{i \in I \mid v \in \chi_i\}$ forms a connected subtree of T.*

To distinguish between vertices of the original graph G and vertices of T in $TD(G)$, we call vertices of T *nodes* and their corresponding χ_i's *bags*. The maximum size of a bag in $TD(G)$ minus one is called the *width* of the tree decomposition. The *treewidth* of a graph G, denoted by $tw(G)$, is the minimum width over all possible tree decompositions of G.

Eppstein [34] introduced the notion of "bounded local treewidth", which is a generalization of the notion of treewidth. A graph has *bounded local treewidth* (or *locally bounded treewidth*) if, for all $r \in \mathbb{N}$, the treewidth of the r-neighborhood of every vertex $v \in V(G)$ is bounded above by a function $f(r)$. Indeed, the bidimensionality of diameter, or more precisely the resulting parameter-treewidth relation, was the point of Eppstein's work (see [25]) in introducing such a class of graphs.

A graph is called an *apex graph* if deleting one vertex produces a planar graph. Eppstein [34] showed that a minor-closed graph class $\mathcal{E}$ has bounded local treewidth if and only if $\mathcal{E}$ is H-minor free for some apex graph H. In particular, he proved that any apex-minor-free class of graphs has at most doubly exponential local treewidth, i.e., $f(r) = 2^{2^{O(r)}}$. See also [25] for a simplified proof and slightly better bounds. It is known that planar graphs, bounded-genus graphs [34], and single-crossing-minor-free graphs [29] have linear local treewidth, i.e., $f(r) = O(r)$. Recently, the authors [26] proved that all apex-minor-free classes of graphs have linear local treewidth. We use this linearity of local treewidth throughout this paper.

A simpler kind of apex graph is a *single-crossing graph*, which can be drawn in the plane with at most one crossing. Single-crossing-minor-free graphs have been studied in [55, 30, 29].

3 Bidimensionality

A series of papers [30, 22, 24, 23] introduce the theory of *bidimensionality* as a general approach for obtaining treewidth-parameter bounds and subexponential fixed-parameter algorithms. This framework is sufficiently broad that an algorithmic designer only needs to check two simple properties of any desired parameter to determine the applicability and practicality of the approach. Indeed, the bidimensionality theory captures essentially all subexponential algorithms obtained so far, and in this paper we show that the theory extends to obtain PTASs as well.

Define the *parameter* corresponding to an optimization problem to be the function mapping graphs to the solution value of the optimization problem; this converts any optimization problem into a parameterized problem. A parameterized problem is *$h(r)$-minor-bidimensional* if the parameter is at least $h(r)$ in an $r \times r$ "grid-like graph" and if the parameter does not increase when taking minors. A parameterized problem is $h(r)$-*contraction-bidimensional* if the parameter is at least $h(r)$ in an $r \times r$ "grid-like graph" and if the parameter does not increase when contracting edges. Our results of course depend on the function $h(r)$. For all bidimensional parameters considered in this paper, $h(r) = \Theta(r^2)$. An example of a different kind of bidimensional parameter is diameter (not interesting from an approximation point of view, but the basis of locally bounded treewidth), which has $h(r) = \Theta(\log r)$ [25].

Treewidth-parameter bounds have been established for all minor-bidimensional problems in H-minor-free graphs for any fixed graph H [24, 23]. In this case, the notion of "grid-like graph" is precisely the regular $r \times r$ square grid. However, contraction-bidimensional problems (such as dominating set) have proved substantially harder. In particular, the largest class of graphs for which a treewidth-parameter bound can be obtained is apex-minor-free graphs instead of general H-minor-free graphs [23]. Such a treewidth-parameter bound has been obtained for all contraction-bidimensional problems in apex-minor-free-graphs [23]. In this case, the notion of "grid-like graph" is an $r \times r$ grid augmented to have, for each vertex, $O(1)$ edges from that vertex to nonboundary vertices. (Here $O(1)$ depends on H.) Unfortunately, this treewidth-parameter bound is large in general: the treewidth is at most $(\sqrt{k})^{O(\sqrt{k})}$ for a $\Theta(r^2)$-bidimensional parameter k. For a fast approximation algorithm, we typically need a bound sublinear in k. For apex-minor-free graphs, such a bound is known only for the special cases of dominating set and vertex cover [26, 24].

The biggest graph classes for which we know a sublinear (indeed, $O(\sqrt{k})$) treewidth-parameter bound for all $\Theta(r^2)$-contraction-bidimensional problems are single-crossing-minor-free graphs and bounded-genus graphs. For single-crossing-minor-free graphs [30, 29] (in particular, planar graphs [22]), the notion of "grid-like graph" is an $r \times r$ grid partially triangulated by additional edges that preserve planarity. For bounded-genus graphs [31], the notion of "grid-like graph" is such a partially triangulated $r \times r$ grid with up to g additional edges ("handles"), where g is the genus of the original graph. (The same result was established for a subset of contraction-bidimensional problems, called α-splittable problems, previously in [24].)

4 Generic PTAS for Minor-Bidimensional Parameters

We consider families of problems in which we are given a graph and our goal is to find a minimum-size set of vertices and/or edges satisfying a certain property. In this section we prove the following result.

THEOREM 4.1. *Consider a $\Theta(r^2)$-minor-bidimensional problem that satisfies the separation property described below. Suppose that the problem can be solved on a graph G with n vertices in $f(n, \mathbf{tw}(G))$ time. Suppose also that the problem can be approximated within a factor of α in $g(n)$ time. Then there is a $(1+\varepsilon)$-approximation algorithm whose running time is $O(nf(n, O(\alpha^2/\varepsilon)) + n^3 g(n))$ for planar and single-crossing-minor-free graphs and $O(nf(n, O(\alpha^2 \lg n/\varepsilon)) + n^3 g(n))$ for bounded-genus graphs.*

More generally, this theorem holds whenever the minor-bidimensional problem has an exact algorithm with running time $f'(n,k)$ where k is the size of the optimal solution. In the theorem we use the bidimensional property that $k = O(\sqrt{\mathbf{tw}(G)})$. Without this property, we would replace the instances of $f(n, \mathbf{tw}(G))$ with $f'(n, \mathbf{tw}(G)^2)$. In general it is harder for $f'(n, O(\alpha^4 \lg^2 n/\varepsilon^2))$ to be polynomial because the typical dependence on k is at least 2^k.

This theorem has several immediate consequences:

COROLLARY 4.1. *Suppose $g(n) = n^{O(1)}$ and $\alpha = O(1)$. If $f(n,w) = n^{w^{O(1)}}$, then we obtain a PTAS for planar and single-crossing-minor-free graphs. If $f(n,w) = 2^{O(w)} n^{O(1)}$, then we obtain a PTAS for bounded-genus graphs. If $f(n,w) = 2^{O(w \lg w)} n^{O(1)}$, then we obtain an almost-PTAS for bounded-genus graphs. If $f(n,w) = h(w) n^{O(1)}$, then we obtain an EPTAS for planar and single-crossing-minor-free graphs.*

COROLLARY 4.2. *Suppose $g(n) = n^{O(1)}$ and $\alpha = O(\lg n)$. If $f(n,w) = 2^{O(w)} n^{O(1)}$, then we obtain a PTAS for planar and single-crossing-minor-free graphs.*

COROLLARY 4.3. *There is an EPTAS for feedback vertex set, face cover, vertex cover, minimum maximal matching, and a series of vertex-removal problems in planar and single-crossing-minor-free graphs. There is an almost-PTAS for all of these problems in bounded-genus graphs. Furthermore, there is a PTAS for vertex cover in apex-minor-free graphs.*

The last result follows from the reduction from vertex cover to dominating set [24, Lemma 5.1] together with the $\sqrt{k}$ bound for dominating set in apex-minor-free graphs [26].

4.1 Separation Property. Our PTAS for minor-bidimensional parameters requires three additional straightforward conditions on the problem, all of which are commonly satisfied. Specifically, for the duration of this section, a problem has the *separation property* if it satisfies the following three conditions:

1. If a graph G has k connected components $G_1, G_2, \ldots, G_k$, then an optimal solution for G is the union of optimal solutions for each connected component G_i.
2. There is a polynomial-time algorithm that, given any graph G, given any vertex cut C whose removal disconnects G into connected components $G_1, G_2, \ldots, G_k$, and given an

optimal solution S_i to each connected component G_i of $G - C$, computes a solution S for G such that the number of vertices and/or edges in S within the induced subgraph $G[C \cup \cup_{i \in I} V(G_i)]$ consisting of C and some connected components of $G - C$ is $\sum_{i \in I} |S_i| \pm O(|C|)$ for any $I \subseteq \{1, 2, \ldots, k\}$. In particular, the total cost of S is at most $\mathrm{OPT}(G - C) + O(|C|)$.

3. Given any graph G, given any vertex cut C, and given an optimal solution OPT to G, for any union G' of some subset of connected components of $G - C$, $|\mathrm{OPT} \cap G'| = |\mathrm{OPT}(G')| \pm O(|C|)$.

Condition 2 states that the extra cost introduced by "merging" the components of $G - C$ along the cut C is $O(|C|)$.

4.2 Algorithm. The algorithm proceeds as follows:

1. Maintain an overall vertex cut C in the original graph; initially $C = \emptyset$.
2. Maintain a set of graphs and their approximate solution costs according to the α-approximation algorithm. Initially, this set consists of just the input graph.
3. For any graph G in this set whose α-approximate solution cost is larger than $b(\varepsilon)$, we cut the graph into two replacement graphs as follows:
   (a) Compute a tree decomposition of G of width w approximately equal to the treewidth $\mathbf{tw}(G)$ of G. For planar graphs [56] and single-crossing-minor-free graphs [29], we obtain a constant-factor approximation: $w = O(\mathbf{tw}(G))$. In general, we obtain a log-OPT approximation [5]: $w = O(\mathbf{tw}(G) \lg \mathbf{tw}(G))$.
   (b) For each node in the tree decomposition, consider the cut formed by the vertices in the corresponding bag. Apply the α-approximation algorithm to each connected component resulting from the cut, and call the approximate solution cost the *weight* of the connected component. Cluster the connected components into two groups by repeatedly placing the heaviest connected component into the lighter group. Among all cuts, choose the one for which the ratio between the weights of the heavier group and lighter group is closest to 1. Add the vertices of this cut to the overall vertex cut C. The two replacement graphs are formed by the two groups.
4. Replace each graph in the set with its connected components. Apply the $f(|H|, \mathbf{tw}(H))$-time algorithm to find the optimal solution to each graph H in the set. Combine these solutions into an approximate solution for the original input graph using the Separation Property.

4.3 Analysis. Before we can analyze the approximation ratio and running time of our algorithm, we need two basic results. These results generalize existing results on separators in low-treewidth graphs [13] to balanced partitions of arbitrary subsets of vertices and/or edges in a low-treewidth graph. Here a tree decomposition gives us extensive additional structure to find such balanced partitions.

LEMMA 4.1. *For any graph G, for any tree decomposition of G of width w, and for any set S of vertices and/or edges, we can remove all ($\leq w+1$) vertices in some bag so that each remaining connected component has at most $|S|/2$ vertices and/or edges from S.*

COROLLARY 4.4. *For any graph G, for any tree decomposition of G of width w, and for any set S of vertices and/or edges, we can remove all ($\leq w+1$) vertices in some bag and cluster the remaining connected components into exactly two groups such that the number of vertices and/or edges from S in each group is at most $(2/3)|S|$.*

Now we proceed to the analysis.

LEMMA 4.2. *Let $\beta > 1/(1+1/(4\alpha^2+\alpha))$ and suppose that $|\mathrm{OPT}(G)|$ is sufficiently large. If Step 3 of the algorithm splits graph G into graphs G_1 and G_2 using cut C, then $|\mathrm{OPT}(G_i)| \leq \beta |\mathrm{OPT}(G)|$ for $i \in \{1, 2\}$.*

Proof. First we bound the ratio between the weights of the heavier group and the lighter group chosen in Step 3. By Corollary 4.4, there is a bag C in the tree decomposition of G whose removal disconnects G into two groups G_1 and G_2 such that $\mathrm{OPT}(G)$ is roughly evenly split between G_1 and G_2. More precisely, if we define $\mathrm{OPT}'_i = \mathrm{OPT}(G) \cap G_i$, then $\frac{1}{2}|\mathrm{OPT}'_2| \leq |\mathrm{OPT}'_1| \leq 2|\mathrm{OPT}'_2|$. Define $\mathrm{OPT}_i = \mathrm{OPT}(G_i)$ and assume by symmetry that $|\mathrm{OPT}_2| \geq |\mathrm{OPT}_1|$. By the Separation Property (3), $|\mathrm{OPT}'_i| = |\mathrm{OPT}_i| \pm \Theta(|C|) = |\mathrm{OPT}_i| \pm \Theta(\sqrt{|\mathrm{OPT}|} \lg |\mathrm{OPT}|)$. Therefore, $\frac{1}{2}|\mathrm{OPT}_2| - O(\sqrt{|\mathrm{OPT}|} \lg |\mathrm{OPT}|) \leq |\mathrm{OPT}_1| \leq 2|\mathrm{OPT}_2| + O(\sqrt{|\mathrm{OPT}|} \lg |\mathrm{OPT}|)$.

By the separation property, $|\mathrm{OPT}| \leq |\mathrm{OPT}_1| + |\mathrm{OPT}_2| + O(\sqrt{|\mathrm{OPT}|} \lg |\mathrm{OPT}|)$, or equivalently $|\mathrm{OPT}| - O(\sqrt{|\mathrm{OPT}|} \lg |\mathrm{OPT}|) \leq |\mathrm{OPT}_1| + |\mathrm{OPT}_2|$. Thus, $|\mathrm{OPT}| - O(\sqrt{|\mathrm{OPT}|} \lg |\mathrm{OPT}|) \leq 2 \max\{|\mathrm{OPT}_1|, |\mathrm{OPT}_2|\} = 2|\mathrm{OPT}_2|$ by our assumption that $|\mathrm{OPT}_2| \geq |\mathrm{OPT}_1|$.

For any $\delta > 0$, if OPT is sufficiently large, $\sqrt{|\mathrm{OPT}|} \lg |\mathrm{OPT}| \leq \delta |\mathrm{OPT}|$ and $(1-\delta)|\mathrm{OPT}| \leq 2|\mathrm{OPT}_2|$. Thus, $\sqrt{|\mathrm{OPT}|} \leq 2\frac{\delta}{1-\delta}|\mathrm{OPT}_2|$. Therefore, for any desired $\delta' > 0$, we can choose δ sufficiently small so that $\frac{1}{2}|\mathrm{OPT}_2| - \delta'|\mathrm{OPT}_2| \leq |\mathrm{OPT}_1| \leq 2|\mathrm{OPT}_2| + \delta'|\mathrm{OPT}_2|$. Because $1/(1/2 - \delta') > 2 + 4\delta' > 2 + \delta'$, $(1/2 - \delta')|\mathrm{OPT}_2| \leq |\mathrm{OPT}_1| \leq \frac{1}{1/2-\delta'}|\mathrm{OPT}_2|$.

The algorithm considers the α-approximate solution APX_i for G_i. Because $|\mathrm{OPT}_i| \leq |\mathrm{APX}_i| \leq \alpha|\mathrm{OPT}_i|$, $\frac{1/2-\delta'}{\alpha}|\mathrm{APX}_2| \leq |\mathrm{APX}_1| \leq \frac{\alpha}{1/2-\delta'}|\mathrm{APX}_2|$. Therefore, APX_i (and hence each connected component of APX_i) has size at most $\lambda(|\mathrm{APX}_1| + |\mathrm{APX}_2|)$ where $\lambda = 1/\left(1 + \frac{1/2-\delta'}{\alpha}\right)$. Repeatedly adding the largest connected component to the smallest group according to Step 3b of the algorithm results in a clustering into two groups G'_1 and G'_2 where $\frac{1-\lambda}{1+\lambda}|\mathrm{APX}(G'_2)| \leq |\mathrm{APX}(G'_1)| \leq \frac{1+\lambda}{1-\lambda}|\mathrm{APX}(G'_2)|$.

The algorithm considers this clustering for bag C, as well as all other bags, and takes the clustering that is most balanced. Therefore the clustering found by the algorithm satisfies the balance property above. Call the two groups in this clustering $\tilde{G}_1$ and $\tilde{G}_2$. Define $\widetilde{\mathrm{APX}}_i$, $\widetilde{\mathrm{OPT}}_i$, and $\widetilde{\mathrm{OPT}}_i'$ as before but with G_i replaced by $\tilde{G}_i$.

Because $|\widetilde{\mathrm{OPT}}_i| \le |\widetilde{\mathrm{APX}}_i| \le \alpha|\widetilde{\mathrm{OPT}}_i|$, $\frac{1-\lambda}{\alpha(1+\lambda)}|\widetilde{\mathrm{OPT}}_2| \le |\widetilde{\mathrm{OPT}}_1| \le \frac{\alpha(1+\lambda)}{1-\lambda}|\widetilde{\mathrm{OPT}}_2|$. By minor bidimensionality and the separation property, $|\widetilde{\mathrm{OPT}}_1| + |\widetilde{\mathrm{OPT}}_2| \le |\mathrm{OPT}|$. Therefore, $|\widetilde{\mathrm{OPT}}_i| \le \frac{1}{1+\frac{1-\lambda}{\alpha(1+\lambda)}}|\mathrm{OPT}| = \frac{1}{1+\frac{1/2-\delta'}{\alpha(2\alpha+1/2-\delta')}}|\mathrm{OPT}|$. Thus we obtain the theorem with $\beta = 1/\left(1+\frac{1/2-\delta'}{\alpha(2\alpha+1/2-\delta')}\right)$. Because $\delta' > 0$ can be chosen arbitrarily small, β can be made arbitrary close to $1/\left(1+\frac{1/2}{\alpha(2\alpha+1/2)}\right)$. □

In the following results we let β denote any number satisfying the condition in Lemma 4.2.

LEMMA 4.3. *The size of the overall vertex cut is $O(|\mathrm{OPT}(G)|/(\sqrt{b(\varepsilon)}(1-\sqrt{\beta})))$ for planar and single-crossing-minor-free graphs, and is $O((|\mathrm{OPT}(G)| \lg |\mathrm{OPT}(G)|)/(\sqrt{b(\varepsilon)}(1-\sqrt{\beta})))$ for bounded-genus graphs.*

Proof. Define the *level* of each graph in the final set in Step 4 to be 0. Define the *level* of each graph that is split in Step 3 to be 1 larger than the maximum level of each of the two pieces resulting from the split. Let $K_1, K_2, \ldots, K_p$ be the graphs at level $\ell \ge 1$. From the algorithm we obtain a vertex cut in the original graph G whose removal leaves a graph G' consisting of exactly $K_1, K_2, \ldots, K_p$ as disconnected pieces. By minor-bidimensionality, $|\mathrm{OPT}(G')| \le |\mathrm{OPT}(G)|$. By the Separation Property, $|\mathrm{OPT}(K_1)| + |\mathrm{OPT}(K_2)| + \cdots + |\mathrm{OPT}(K_p)| \le |\mathrm{OPT}(G')| \le |\mathrm{OPT}(G)|$. By Lemma 4.2, $|\mathrm{OPT}(K_i)| \ge b(\varepsilon)/\beta^{\ell-1}$. Therefore, $p \le |\mathrm{OPT}(G)|\beta^{\ell-1}/b(\varepsilon)$.

The size of the cut introduced in Step 3 for splitting K_i is $w(K_i)+1$, which is $O(\sqrt{|\mathrm{OPT}(K_i)|}\lg|\mathrm{OPT}(K_i)|)$. The total cut size over all K_i's is $O(\sum_{i=1}^{p}\sqrt{|\mathrm{OPT}(K_i)|}\lg|\mathrm{OPT}(K_i)|)$, which is at most $O(\sum_{i=1}^{p}\sqrt{|\mathrm{OPT}(K_i)|}\lg|\mathrm{OPT}|)$. This sum is maximized when $|\mathrm{OPT}(K_i)| = |\mathrm{OPT}(G)|/p$. Thus the total cut size at level ℓ is $O(\sqrt{p}\sqrt{|\mathrm{OPT}(G)|}\lg|\mathrm{OPT}(G)|) = O(|\mathrm{OPT}(G)|\lg|\mathrm{OPT}(G)| \cdot \beta^{(\ell-1)/2}/\sqrt{b(\varepsilon)})$. Therefore, the total cut size is $O\left(\sum_{\ell=1}^{\infty}|\mathrm{OPT}(G)|\lg|\mathrm{OPT}(G)| \cdot \beta^{(\ell-1)/2}/\sqrt{b(\varepsilon)}\right) = O\left(\frac{|\mathrm{OPT}(G)|\lg|\mathrm{OPT}(G)|}{\sqrt{b(\varepsilon)}(1-\sqrt{\beta})}\right)$. For planar and single-crossing-minor-free graphs, we avoid the $\lg|\mathrm{OPT}(K_i)|$ factor and thus the $\lg|\mathrm{OPT}(G)|$ factor. □

THEOREM 4.2. *The approximate solution produced by the algorithm is within a factor of $1+\varepsilon$ times optimal if we set $b(\varepsilon) = \Omega(1/(\varepsilon^2(1-\sqrt{\beta})^2))$ for planar and single-crossing-minor-free graphs and $b(\varepsilon) = \Omega((\lg^2 n)/(\varepsilon^2(1-\sqrt{\beta})^2))$ for general graphs.*

Proof. We concentrate on the case of general graphs; planar and single-crossing-minor-free graphs simply omit the log factor. Lemma 4.3 bounds the error by $O((|\mathrm{OPT}(G)|\lg|\mathrm{OPT}(G)|)/(\sqrt{b(\varepsilon)}(1-\sqrt{\beta})))$. The PTAS needs this error to be at most $\varepsilon|\mathrm{OPT}(G)|$. This bound is guaranteed to hold if $b(\varepsilon) \ge \frac{(2\lg|\mathrm{OPT}(G)|+1)^2}{\varepsilon^2(1-\sqrt{\beta})^2}$. Applying the Separation Property with $C = V(G)$, we know that $|\mathrm{OPT}(G)| = O(n)$. Therefore it suffices to set $b(\varepsilon) \ge \Omega\left(\frac{\lg^2 n}{\varepsilon^2(1-\sqrt{\beta})^2}\right)$. □

COROLLARY 4.5. *If we set $b(\varepsilon) = \Theta(1/(\varepsilon^2(1-\sqrt{\beta})^2))$, then the running time of the $(1+\varepsilon)$-approximation algorithm is $O(nf(n, O(\alpha^2/\varepsilon)) + n^3 g(n))$ for planar and single-crossing-minor-free graphs and $O(nf(n, O(\alpha^2 \lg n/\varepsilon)) + n^3 g(n))$ for bounded-genus graphs.*

Proof. A simple asymptotic analysis shows that $1/(1-\sqrt{\beta}) \sim 8\alpha^2$ and thus $b(\varepsilon) \sim 64\alpha^4/\varepsilon^2$. Therefore the size of the optimal solution for every graph in the final set (in Step 4) is $O(\alpha^4/\varepsilon^2)$. By bidimensionality, the treewidth of these graphs is $O(\alpha^2/\varepsilon)$. There are $O(n)$ such graphs, so we make $O(n)$ calls to the exact algorithm with running time $f(n, O(\alpha^2/\varepsilon))$. In Step 3b we make $O(n^2)$ calls to the α-approximation algorithm, $O(n)$ for each candidate cut. Step 3 iterates $O(n)$ times, so we make $O(n^3)$ calls to the approximation algorithm with running time $g(n)$. □

This result proves Theorem 4.1.

5 Generic PTAS for Contraction-Bidimensional Parameters

Consider a problem P where the input is a graph and the output is a minimum-size set S of vertices and/or edges with a certain property π. The *generalized form* of such a problem P is another problem where the input is a graph and a set C of vertices and the output is a minimum-size set S of vertices and/or edges such that $S \cup C \cup E(G[C])$ satisfies property π. The *cost* of such a solution is the size of S, not counting the size of $C \cup E(G[C])$. (Thus an α-approximation algorithm for the generalized form of problem P is an algorithm whose output S is within an α factor of the optimal size of S.) In other words, we get vertices in C and edges in $E(G[C])$ "for free" (assuming their addition helps to satisfy π).

THEOREM 5.1. *Consider a contraction-bidimensional problem that satisfies the separation property described below. Suppose that the generalized problem can be solved on a graph G with n vertices in $f(n, \mathbf{tw}(G))$ time. Suppose also that the generalized problem can be approximated within a factor of α in $g(n)$ time. Then there is a $(1+\varepsilon)$-approximation algorithm whose running time is $O(nf(n, O(\alpha^2/\varepsilon)) + n^3 g(n))$ for planar and single-crossing-minor-free graphs and $O(nf(n, O(\alpha^2 \lg n/\varepsilon)) + n^3 g(n))$ for bounded-genus graphs (or any graph class where the parameter satisfies the $O(\sqrt{k})$ parameter-treewidth bound).*

Corollaries 4.1 and 4.2 therefore also generalize to the contraction-bidimensional case under the additional assumptions stated in Theorem 5.1. In addition, we obtain the following corollary about specific problems:

COROLLARY 5.1. *There is a PTAS for dominating set, edge dominating set, r-dominating set, and clique-transversal set in apex-minor-free graphs. There is a PTAS for connected dominating set, connected edge dominating set, and connected r-dominating set in planar and single-crossing-minor-free graphs; and almost-PTASs for the same problems in apex-minor-free graphs. Furthermore, all of these PTASs are EPTASs for planar graphs.*

Here we use that dominating set, and therefore any parameter bounded above by dominating set, satisfies the $\sqrt{k}$ bound for apex-minor-free graphs [26].

5.1 Separation Property. For contraction-bidimensional parameters, the exact requirements on the problem are slightly different but similarly straightforward. The main distinction is that the connected components are always considered together with the cut C. Specifically, for the duration of this section, a problem has the *separation property* if it satisfies the following two conditions:

1. There is a polynomial-time algorithm that, (a) given any graph G, (b) given any vertex set C, (c) given a set of graphs $\{G_1, G_2, \ldots, G_r\}$ such that $\{V(G_i) - C \mid 1 \leq i \leq r\}$ partitions $V(G) - C$, and (d) given a solution S_i to the generalized problem for each graph G_i with vertex set $C \cap V(G_i)$, computes a solution S to the original problem for G such that the number of vertices and/or edges in S within any union $G' = \cup_{i \in I} G_i$ of some of the G_i's is $\sum_{i \in I} |S_i| \pm O(|C|)$ for any $I \subseteq \{1, 2, \ldots, r\}$. In particular, the total cost of S is at most $\sum_{i=1}^{r} |S_i| + O(|C|)$.
2. Given (a) any graph G, (b) any vertex set C, (c) a set of graphs $\{G_1, G_2, \ldots, G_r\}$ such that $\{V(G_i) - C \mid 1 \leq i \leq r\}$ partitions $V(G) - C$, and (d) an optimal solution OPT to G, any union $G' = \cup_{i \in I} G_i$ of some of the G_i's, $I \subseteq \{1, 2, \ldots, r\}$ satisfies $|\mathrm{OPT} \cap G'| = |\mathrm{OPT}(G')| \pm O(|C|)$.

5.2 Algorithm. The algorithm proceeds as follows:

1. Maintain an overall vertex cut C in the original graph. Initially $C = \emptyset$.
2. Maintain a set of graphs $\{G_1, G_2, \ldots, G_r\}$. Initially, this set consists of just the input graph G.
3. Maintain, for each $i \in \{1, 2, \ldots, r\}$, the α-approximate solution cost to the generalized problem involving graph G_i and vertex set $C_i = C \cap V(G_i)$.
4. For any graph G_i in this set whose α-approximate solution cost is larger than $b(\varepsilon)$ and whose α-approximate solution set S is at most as large as $C_i \cup E(G[C_i])$, we cut the graph into two replacement graphs as follows:
   (a) Compute a tree decomposition of G_i of width w approximately equal to the treewidth $\mathbf{tw}(G_i)$ of G_i. For planar graphs [56] and single-crossing-minor-free graphs [29], we obtain a constant-factor approximation: $w = O(\mathbf{tw}(G_i))$. In general, we obtain a log-OPT approximation [5]: $w = O(\mathbf{tw}(G_i) \lg \mathbf{tw}(G_i))$.
   (b) For each node in the tree decomposition, consider the cut C' formed by the vertices in the corresponding bag. For each connected component X resulting from the cut, apply the α-approximation algorithm to the graph $G[X \cup C']$ with vertex set $C' \cup (C_i \cap X)$, and call the approximate solution cost the *weight* of the connected component X. Cluster the connected components into two groups by repeatedly placing the heaviest connected component into the lighter group. Among all cuts, choose the one for which the ratio between the weights of the heavier group and lighter group is closest to 1. Add the vertices of this cut C' to the overall vertex cut C. For each of the two groups, we take the union Y of all connected components in the group and form the graph $G_i[Y \cup C']$. The two graphs resulting from the two groups are the replacement graphs for G_i.
5. For graph G_i in the set whose α-approximate solution cost is at most $b(\varepsilon)$, find the optimal solution to the generalized problem with graph G_i and vertex set C_i using a $2^{O(\mathbf{tw}(G_i))} n^{O(1)}$ fixed-parameter algorithm. (For planar and single-crossing-minor-free graphs, we can in fact use any fixed-parameter algorithm for the treewidth parameter.) For graphs G_i in the set whose α-approximate solution set S is larger than $C_i \cup E(G[C_i])$, we use the existing α-approximate solution.
6. Combine these solutions into an approximate solution for the original input graph using the Separation Property.

Analysis sketch. The main difference in the analysis, compared to the minor-bidimensional case of Section 4, is that for some of the graphs G_i in the final set in Step 5 we use approximate solutions instead of exact solutions. This approximation happens only when the vertex set C_i becomes too large. In this case we charge the excess cost from the approximate solution to the nodes in that vertex cut C_i. We argue that each node of C gets charged to at most twice, once on each side of the recursion where that node of C is split.

A smaller difference is that, as we split, we increase the sums of the sizes of the optimal solutions among the graphs in the set, because of the duplication of nodes in the vertex set C. As a result, most bounds gain lower-order terms. These terms can be compensated by enlarging $b(\varepsilon)$ slightly.

6 APTAS for Graphs of Locally Bounded Treewidth

The main result of this section is as follows.

THEOREM 6.1. *For any $\varepsilon > 0$, the minimum connected dominating set problem on minor-closed graphs of locally bounded treewidth has an approximation scheme with approximation ratio $1 + \varepsilon$ and running time $n^{O((1/\varepsilon) \lg(1/\varepsilon) \lg \lg n)}$.*

The proof of this theorem captures the main ideas of our extension of Baker's approach to nonlocal properties like connected dominating set. In particular, the same approach can be used to obtain analogous results for connected vertex cover, connected edge-dominating set, and connected r-dominating set. In Section 7, we show how we can reduce the running time from $n^{O(\lg \lg n)}$ to $n^{O(1)}$, i.e., obtaining a PTAS, on planar graphs. We conjecture that in fact the same trick can be applied to obtain a PTAS for apex-minor-free graphs.

The following generalization of connected dominating set plays an important role in our algorithms.

DEFINITION 6.1. *The* generalized connected dominating set (GCDS) *problem is defined as follows. Given a graph G and a set $I \subseteq V(G)$ called* the interior*, determine a subset D of $V(G)$ of minimum size with the property that, for every connected component C of I, the dominating vertices in or neighboring C, $D \cap (C \cup N(C))$, dominate C and belong to one connected component of $G[D]$.*

In particular, if we set $I = V(G)$, then the GCDS problem is the same as the connected dominating set problem.

In Section 6.1, we describe the APTAS algorithm except for one dynamic programming subroutine. In Section 6.2, we prove correctness and analyze the approximation ratio. We require a dynamic programming subroutine for GCDS on graphs of bounded treewidth given by the following theorem:

THEOREM 6.2. *The GCDS problem for given graph G and set I can be solved in time $O(w^w \cdot |V(G)|)$ when a tree decomposition of width w for G is given.*

We omit the proof of this theorem because of lack of space in this extended abstract. Even this theorem solves an open problem of [1]. Previously it was not believed that such nonlocal properties as connected dominating set could be captured by bounded-treewidth dynamic programs.

6.1 Algorithm. In this section, we present the APTAS algorithm for connected dominating set in a minor-closed class of graphs of locally bounded treewidth. As a starting point, we consider a simple constant-factor approximation for connected dominating set:

THEOREM 6.3. *For any $\delta > 0$, there is a $(3 + \delta)$-approximation algorithm for the connected dominating set problem on minor-closed graphs of locally bounded treewidth.*

Proof. Using the algorithm of Eppstein [34] or Demaine and Hajiaghayi [26], we know that dominating set has a PTAS on minor-closed graphs of locally bounded treewidth. Now, one can easily observe that for any dominating set D in a connected graph G, we can add at most $2|D| - 2$ vertices to make D connected (by adding two vertices we can decrease the number of connected components by one). Thus we obtain a connected dominating set whose size is $(1+\delta')OPT + 2(1+\delta')OPT - 2$ where OPT is the size of a minimum dominating set and thus a lower bound on the size of a minimum connected dominating set. The result follows immidiately by taking $\delta' = \delta/3$. □

We are now ready to describe the algorithm.

First we compute the $(3 + \delta)$-approximate solution B from Theorem 6.3. Let $k = \frac{1}{(16+4\delta)\varepsilon}(\lg n + 1)$. We assume that ε is small enough so that $k \geq 4(\lg n + 1)$; otherwise, the $(3 + \delta)$-approximate solution B gives the desired approximation factor.

Next we decompose the vertex set of G into vertex sets such that the subgraph induced on each set has small (logarithmic) treewidth. In a breadth-first search tree from an arbitrary vertex $v \in V(G)$, let L_h denote the set of vertices at *layer* (or *level* or *distance*) h in the tree. Also let $L[e, f] = L_e \cup L_{e+1} \cup \cdots L_f$ denote several consecutive layers. Then the vertex sets in our decomposition are as follows: for $1 \leq i \leq k$ and $j \geq 0$, we define $L_{ij} = L[(j-1)k + i - 2(\lg n + 1), jk + i + 2(\lg n + 1) - 1]$.

By the results of Demaine and Hajiaghayi [26], the treewidth of any r consecutive layers in a graph from a minor-closed class of graphs of locally bounded treewidth is at most $cr + d$ for some constants c and d. Thus, the treewidth of $G[L_{ij}]$ is at most $c(k + 4(\lg n + 1)) + d$. Using an algorithm of Amir [5], we construct a tree decomposition of width at most $\frac{11}{3}(c(k + 4(\lg n + 1)) + d)$ for each $G[L_{ij}]$ in $O(2^{3.698(c(k+4(\lg n+1))+d)} n^{3+\varepsilon})$ time. Note that we could not use existing exact tree-decomposition algorithms [12] because the running time would be too high.

We solve a GCDS instance on each L_{ij} with the interior defined as the set $I_{ij} = L[(j-1)k+i, jk+i-1]$. In contrast to Baker's approach, here the number of layers (thickness) outside the interior is $\Theta(\lg n)$. This aspect plays a crucial role in Lemma 6.1. By Theorem 6.2, we can compute the optimal solutions Opt_{ij} for each instance $G[L_{ij}], I_{ij}$ in $O((k + 2(\lg n + 1))^{k+2(\lg n+1)}|V(G[L_{ij}])|)$ time.

Let $Opt_i = \cup_{j \geq 0} Opt_{ij} \cup B_i$, where $B_i = \cup_{j \geq 0} B_{ij}$ and $B_{ij} = B \cap (L[(j-1)k + i, (j-1)k + i + 1] \cup L[jk + i - 1, jk + i])$. Here we see another of the main differences from Baker's approach: the auxiliary approximate solution B serves as a "backbone" to connect adjacent Opt_{ij}'s. Because, for fixed i, each vertex of G appears in at most two L_{ij}'s, computing each Opt_i takes $O((k + 2(\lg n + 1))^{k+2(\lg n+1)} n)$ time.

We take Opt_m to be the solution of minimum weight among $Opt_1, Opt_2, \ldots, Opt_k$ as our solution on graph G. In the next subsection, we show that it has at most a ratio $1 + \varepsilon$ of the optimal. A time bound of $n^{O((1/\varepsilon) \lg(1/\varepsilon) \lg \lg n)}$ follows immediately from the time needed to construct the tree decompositions, the number of Opt_i's, and the time to compute each of them.

6.2 Correctness and Approximation Factor. First, we show the correctness of the algorithm. By the properties of Opt_{ij}'s, Opt_i is a dominating set for G (for fixed i, each vertex appears once in some interior set I_{ij} and thus dominated by at least one vertex). This means Opt_m is a dominating set for graph G. Next, we consider the connectivity of Opt_m. For any j and for any connected component C of the interior of L_{mj}, if a connected component C' of B_m has a vertex in C, then C' must connect to the connected component D_C for C. In this way D_C makes up the connections of backbone B lost from cutting a part in C. Thus, the connectivity of Opt_m follows from the connectivity of the backbone B. This is the main part where we use the connectivity of backbone B.

We now compare the size of Opt_m with respect to a global OPT, i.e., a minimum connected dominating set over the whole graph G. More precisely, we show that $\frac{|Opt_m|}{|OPT|} \le 1 + \varepsilon$.

Before starting the proof, we need the following facts and the following important lemma.

FACT 6.1. *Each vertex of B appears in at most four Opt_i, $1 \le i \le k$.*

FACT 6.2. *For fixed i, L_{ij} and $L_{i(j+1)}$ intersect only in two consecutive layers. However L_{ij} and $L_{i(j+2)}$ do not intersect, because $k > 4(\lg n + 1)$ (see the statement of Theorem 6.1). This implies that each vertex appears in exactly $k + 4(\lg n + 1)$ sets L_{ij}.*

LEMMA 6.1. *Let OPT be an optimal solution to connected dominating set in the whole graph G, let $L_{ij} = L[(j-1)k + i-2(\lg n+1), jk+i+2(\lg n+1)-1]$ be $k+4(\lg n+1)$ layers of this graph, and let Opt_{ij} be the solution to the GCDS dynamic program in layer L_{ij} with interior $I_{ij} = L[(j-1)k + i, jk + i - 1]$. Then we have $|OPT \cap L_{ij}| \ge |Opt_{ij}|$.*

Proof. We claim that we can reduce $OPT \cap L_{ij}$ to a solution to GCDS in $G[L_{ij}]$ without increase in size. This claim immediately implies the lemma because Opt_{ij} is a solution of minimum size to GCDS. Consider the intersection $OPT_{ij} = OPT \cap I'_{ij}$ where $I'_{ij} = L[(j-1)k+i-1, jk+i]$ consists of I_{ij} plus one additional layer on each side. We easily observe that OPT_{ij} is a dominating set for I_{ij}. The only issue is that some vertices in OPT_{ij} are not connected but need to be for a solution to GCDS.

To abstract the connectivity requirements, we define a graph H as follows. Start with the subgraph of G induced by $OPT - I'_{ij}$. For each connected component C of OPT_{ij}, we add a vertex C to H and connect it to every vertex v of H that has a neighbor in connected component C in the original graph G. We call C a *connected component (CC)* vertex; intuitively, it represents the contraction of vertices in a connected component of OPT_{ij}. Because OPT is a connected dominating set, H is connected.

To fix the connectivity requirements, we maintain a forest in H, consisting initially of just a spanning tree of H. We repeatedly perform one of the following two modifications to a tree T in the current forest. First, we try to remove from T any vertex v that leaves connected all CC vertices that should be connected in a GCDS solution. This operation splits T into possibly several trees in the forest. Second, if there exist two degree-two vertices u and v in $L_{ij} - I'_{ij}$ connected by an edge in T, then we remove both u and v from T. As a result of this removal, we obtain exactly two groups of CC vertices that are disconnected in T but some pairs must be connected for a solution to GCDS. If we connect any two CC vertices from the two groups, we will re-obtain the desired connectivity. Any pair of CC vertices that must be connected for a solution to GCDS must belong to a common connected component of $G[I'_{ij}]$ (by definition of GCDS). Therefore, if we start from such a pair of CC vertices and explore neighboring CC vertices, which all come from I'_{ij} and thus the same connected component of $G[I'_{ij}]$, we eventually obtain two CC vertices whose corresponding connected components in G are at distance at most three (because OPT_{ij} is a dominating set for I_{ij}). Hence, there are at most two vertices in a connected component of a I_{ij} which connect these two connected components. We add these two vertices to T and the resulting connections, restoring the property that T is a tree. Because these new vertices do not belong to $L_{ij} - I'_{ij}$, they will not be removed as degree-two vertices in the future.

By the end of this process, we have connected any two CC vertices that should be connected for a solution to GCDS, without having increased the total number of vertices of the forest. However, these connections may use vertices outside L_{ij}, but a solution to GCDS must be contained in L_{ij}. We claim that, in fact, all vertices in the forest are inside L_{ij}. Define the *height* of a vertex to be the number of edges along a shortest path to a CC vertex (roughly corresponding to which layer surrounding I'_{ij} contains the vertex). Thus we view CC vertices as *leaves*. Along any path from a leaf to a vertex of height h, at least $\lfloor h/2 \rfloor$ of the vertices must have degree at least three. Hence, the number of leaves below a node of height h must be at least $2^{\lfloor h/2 \rfloor}$. Because there are at most n leaves, the height of any node is at most $2 \lg n + 1$. Therefore, every vertex in the forest is contained in L_{ij}, so we obtain a solution to GCDS of the desired size. □

Roughly speaking, Lemma 6.1 implies that the thickness that we have considered for boundary layers in L_{ij} is enough to get connected all connected components of dominating vertices in its interior.

Using Lemma 6.1 and Facts 6.1 and 6.2, we have $k \cdot |Opt_m| \le^{[\text{Fact 6.1}]} \sum_{i=1}^{k} |Opt_i| + 4|B| \le \sum_{i=1}^{k} \sum_{j \ge 0} |Opt_{ij}| + (12 + 4\delta)|OPT| \le^{[\text{Lemma 6.1}]} \sum_{i=1}^{k} \sum_{j \ge 0} |OPT \cap L_{ij}| + (12 + 4\delta)|OPT| =^{[\text{Fact 6.2}]} (k + 4(\lg n + 1) + 12 + 4\delta) \cdot |OPT|$. Recalling that $k = \frac{1}{(16+4\delta)\varepsilon}(\lg n + 1)$, we have $|Opt_m| \le \frac{k+4(\lg n+1)+12+4\delta}{k}|OPT| \le \left(1 + \frac{(4+12+4\delta)(\lg n+1)}{(\lg n+1)/[(16+4\delta)\varepsilon]}\right)|OPT| = (1 + \varepsilon)|OPT|$, as desired.

7 PTAS for Planar Graphs

In this section we prove the following strengthening of Theorem 6.1 for planar graphs:

THEOREM 7.1. *For any $\varepsilon > 0$, the minimum connected dominating set problem on planar graphs has a polynomial-time approximation scheme that achieves an approximation ratio $1 + \varepsilon$ in $n^{O(1/\varepsilon)}$ time.*

At a high level, our PTAS for planar graphs has only two differences compared to the APTAS for graphs of bounded local treewidth from Section 6. First, we change the layering of the graph from breadth-first layers to the "outerplanar layering" of Baker's original paper [8]. Second, we replace the dynamic program for graphs of bounded treewidth with a new dynamic program for graphs of bounded "outerplanarity" (a stronger condition than bounded treewidth). The main difference is that the dynamic program must exploit the planarity of the original graph and the bounded outerplanarity of the subgraph given to the dynamic program, so that its running time is $2^{O(w)}n^{O(1)}$ instead of $w^{O(w)}n^{O(1)}$. The basic idea is that we encode the second and third coordinates of colors more efficiently using the planar structure of the graph, leading to Catalan structures (which have cardinality $2^{O(w)}$) instead of general partition structures (which have cardinality $w^{O(w)}$).

In the absence of space, we omit the formal proof in this extended abstract.

8 Discussion

In this paper we have extended the separation approach and Baker's approach for designing PTASs in planar graphs and their generalizations. Our generalized techniques substantially extend the family of problems for which we can obtain such PTASs. This family of problems closely matches the state-of-the-art of what problems have subexponential fixed-parameter algorithms, providing a strong connection between fixed-parameter tractability and approximation algorithms. Also our view illustrates that the two PTAS techniques are in fact closely linked through bidimensionality.

It would be interesting to see to what extent Baker's approach can be used to obtain PTASs for other "nonlocal" problems on planar graphs such as feedback vertex set. We conjecture that the weighted connected dominating set problem on planar graphs has a PTAS via the generalized Baker's approach. (In general, Baker layerwise decomposition seems to handle weights better than separators or their extensions.) We also believe that our techniques can be combined with the techniques of Grohe [43] to further generalize our PTASs to H-minor-free graphs.

Another consequence of our efficient bounded-treewidth algorithms for connected dominating set is a subexponential algorithm, which was implicitly asked for by Alber et al. [1]. (Note that a bounded-treewidth algorithm also follows from expressing the problem in monadic second-order logic [15], but the dependence on treewidth is highly super-exponential and therefore not fast enough for a subexponential algorithm or a PTAS.)

THEOREM 8.1. *We can decide in* $2^{O(\sqrt{k})}n^{O(1)}$ *(*$2^{O(\sqrt{k}\lg k)}n^{O(1)}$*) time whether a planar graph (apex-minor-free graph) has a connected dominating set of size at most* k.

Addendum. Recently, the bidimensionality theory, in particular the treewidth-parameter bounds described in Section 3, have been improved [27]. Specifically, in [28] it is shown that every $\Theta(r^2)$-minor-bidimensional parameter k in H-minor-free graphs, and every $\Theta(r^2)$-contraction-bidimensional parameter k in apex-minor-free graphs, satisfy a treewidth-parameter bound of $\mathbf{tw}(G) = O(\sqrt{k})$. This result allows us to extend our approximation algorithms for bounded-genus graphs from Theorems 4.1 and 5.1 to H-minor-free and apex-minor-free graphs, respectively.

Another recent result is an $O(1)$-approximation algorithm for treewidth in H-minor-free graphs [35]. Using this algorithm instead of the $O(\lg \mathrm{OPT})$-approximation of [5], we avoid the $\lg n$ term in the bounds of Theorem 4.1 and 5.1. Thus we extend our bounds and PTASs for planar and single-crossing-minor-free graphs to apply to H-minor-free graphs for $\Theta(r^2)$-minor-bidimensional problems and to apex-minor-free graphs for $\Theta(r^2)$-contraction-bidimensional problems. This PTAS result is the most general one could hope for in the context of bidimensionality.

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 (2002), 461–493.

[2] J. ALBER, H. FAN, M. R. FELLOWS, H. FERNAU, R. NIEDERMEIER, F. A. ROSAMOND, AND U. STEGE, *Refined search tree technique for DOMINATING SET on planar graphs*, in Proc. 26th International Symp. Mathematical Foundations of Computer Science, LNCS 2136, 2001, 111–122.

[3] J. ALBER, H. FERNAU, AND R. NIEDERMEIER, *Parameterized complexity: exponential speed-up for planar graph problems*, Journal of Algorithms, 52 (2004), 26–56.

[4] K. ALZOUBI, P. J. WAN, AND O. FRIEDER, *New distributed algorithm for connected dominating set in wireless ad hoc networks*, in Proc. of HICSS, 2002, 3881–3887.

[5] E. AMIR, *Efficient approximation for triangulation of minimum treewidth*, in Proc. 17th Conference on Uncertainty in Artificial Intelligence, 2001, 7–15.

[6] S. ARORA, M. GRIGNI, D. KARGER, P. KLEIN, AND A. WOLOSZYN, *A polynomial-time approximation scheme for weighted planar graph TSP*, in Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms, 1998, 33–41.

[7] V. BAFNA, P. BERMAN, AND T. FUJITO, *A 2-approximation algorithm for the undirected feedback vertex set problem*, SIAM Journal on Discrete Mathematics, 12 (1999), 289–297.

[8] B. S. BAKER, *Approximation algorithms for NP-complete problems on planar graphs*, Journal of the Association for Computing Machinery, 41 (1994), 153–180.

[9] R. BAR-YEHUDA, D. GEIGER, J. NAOR, AND R. M. ROTH, *Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and bayesian inference*, in Proc. 5th Annual ACM-SIAM Symposium on Discrete Algorithms, 1994, 344–354.

[10] C. BAZGAN, *Approximation schemes and parameterized complexity*, PhD thesis, INRIA, Orsay, France, 1995.

[11] A. BECKER AND D. GEIGER, *Approximation algorithms for the loop cutset problem*, in Proc. 10th Annual Conference on Uncertainty in Artificial Intelligence, 1994, 60–68.

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

[13] ———, *A partial k-arboretum of graphs with bounded treewidth*, Theoretical Computer Science, 209 (1998), 1–45.

[14] J. A. BONDY AND U. S. R. MURTY, *Graph Theory with Applications*, American Elsevier Publishing Co., Inc., New York, 1976.

[15] R. B. BORIE, R. G. PARKER, AND C. A. TOVEY, *Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families*, Algorithmica, 7 (1992), 555–581.

[16] L. CAI AND J. CHEN, *On fixed-parameter tractability and approximability of NP optimization problems*, Journal of Computer and System Sciences, 54 (1997), 465–474.

[17] M. CESATI AND L. TREVISAN, *On the efficiency of polynomial time approximation schemes*, Information Processing Letters, 64 (1997), 165–171.

[18] M.-S. CHANG, T. KLOKS, AND C.-M. LEE, *Maximum clique transversals*, in Proc. 27th Internat. Worksh. Graph-Theoretic Concepts in Computer Science, LNCS 2204, 32–43.

[19] X. CHENG, X. HUANG, D. LI, W. WU, AND D.-Z. DU, *A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks*, Networks, 42 (2003), 202–208.

[20] N. CHIBA, T. NISHIZEKI, AND N. SAITO, *An approximation algorithm for the maximum independent set problem on planar graphs*, SIAM Journal on Computing, 11 (1982), 663–675.

[21] F. A. CHUDAK, M. X. GOEMANS, D. S. HOCHBAUM, AND D. P. WILLIAMSON, *A primal-dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs*, Oper. Res. Lett., 22 (1998), 111–118.

[22] E. D. DEMAINE, F. V. FOMIN, M. HAJIAGHAYI, AND D. M. THILIKOS, *Fixed-parameter algorithms for the (k, r)-center in planar graphs and map graphs*, ACM Transactions on Algorithms. To appear.

[23] ———, *Bidimensional parameters and local treewidth*, SIAM Journal on Discrete Mathematics, (2004). To appear.

[24] ———, *Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs*, in Proc. 15th ACM-SIAM Symp. Discrete Algorithms, 2004, 823–832.

[25] E. D. DEMAINE AND M. HAJIAGHAYI, *Diameter and treewidth in minor-closed graph families, revisited*, Algorithmica, 40 (2004), 211–215.

[26] ———, *Equivalence of local treewidth and linear local treewidth and its algorithmic applications*, in Proc. 15th ACM-SIAM Symposium on Discrete Algorithms, 2004, 833–842.

[27] ———, *Fast algorithms for hard graph problems: Bidimensionality, minors, and local treewidth*, in Proc. 12th International Symposium on Graph Drawing, 2004. To appear.

[28] ———, *Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality*, in Proc. 16th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005. To appear.

[29] E. D. DEMAINE, M. HAJIAGHAYI, N. NISHIMURA, P. RAGDE, AND D. M. THILIKOS, *Approximation algorithms for classes of graphs excluding single-crossing graphs as minors*, Journal of Computer and System Sciences, 69 (2004), 166–195.

[30] E. D. DEMAINE, M. HAJIAGHAYI, AND D. M. THILIKOS, *Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors*, Algorithmica. To appear.

[31] ———, *The bidimensional theory of bounded-genus graphs*, in Proc. 29th International Symposium on Mathematical Foundations of Computer Science, 2004, 191–203.

[32] R. G. DOWNEY AND M. R. FELLOWS, *Parameterized Complexity*, Springer-Verlag, New York, 1999.

[33] D. EPPSTEIN, *Subgraph isomorphism in planar graphs and related problems*, in Proc. 6th Annual ACM-SIAM Symposium on Discrete Algorithms, 1995, 632–640.

[34] ———, *Diameter and treewidth in minor-closed graph families*, Algorithmica, 27 (2000), 275–291.

[35] U. FEIGE, M. HAJIAGHAYI, AND J. R. LEE, *Improved approximation algorithms for minimum vertex separator*. Manuscript, 2004.

[36] F. V. FOMIN AND D. M. THILIKOS, *Dominating sets in planar graphs: Branch-width and exponential speed-up*, in Proc. 14th Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, 168–177.

[37] M. FRICK AND M. GROHE, *Deciding first-order properties of locally tree-decomposable structures*, Journal of the ACM, 48 (2001), 1184–1206.

[38] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Co., San Francisco, Calif., 1979.

[39] M. X. GOEMANS AND D. P. WILLIAMSON, *Primal-dual approximation algorithms for feedback problems in planar graphs*, Combinatorica, 18 (1998), 37–59.

[40] M. GRIGNI, *Approximate TSP in graphs with forbidden minors*, in Proc. 27th International Colloquium of Automata, Languages and Programming, LNCS 1853, 2000, 869–877.

[41] M. GRIGNI, E. KOUTSOUPIAS, AND C. PAPADIMITRIOU, *An approximation scheme for planar graph TSP*, in Proc. 36th Symp. Foundations of Computer Science, 1995, 640–645.

[42] M. GRIGNI AND P. SISSOKHO, *Light spanners and approximate TSP in weighted graphs with forbidden minors*, in Proc. 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002, 852–857.

[43] M. GROHE, *Local tree-width, excluded minors, and approximation algorithms*, Combinatorica, 23 (2003), 613–632.

[44] S. GUHA AND S. KHULLER, *Approximation algorithms for connected dominating sets*, Algorithmica, 20 (1998), 374–387.

[45] A. GUPTA, A. KUMAR, M. PÁL, AND T. ROUGHGARDEN, *Approximation via cost-sharing: A simple approximation algorithm for the multicommodity rent-or-buy problem*, in Proc. 44th Symp. Foundations of Computer Science, 2003, 606–615.

[46] G. GUTIN, T. KLOKS, AND C. M. LEE, *Kernels in planar digraphs*, in Optimization Online, Mathematical Programming Society, Philadelphia, 2001.

[47] I. KANJ AND L. PERKOVIĆ, *Improved parameterized algorithms for planar dominating set*, in Proc. 27th International Symposium on Mathematical Foundations of Computer Science, LNCS 2420, 2002, 399–410.

[48] D. R. KARGER AND M. MINKOFF, *Building steiner trees with incomplete global knowledge*, in Proc. 41st Annual Symposium on Foundations of Computer Science, 2000, 613–622.

[49] S. KHANNA AND R. MOTWANI, *Towards a syntactic characterization of PTAS*, in Proc. 28th Annual ACM Symposium on the Theory of Computing, 1996, 329–337.

[50] T. KLOKS, C. M. LEE, AND J. LIU, *New algorithms for k-face cover, k-feedback vertex set, and k-disjoint set on plane and planar graphs*, in Proc. 28th International Workshop on Graph-Theoretic Concepts in Computer Science, LNCS 2573, 2002, 282–295.

[51] R. J. LIPTON AND R. E. TARJAN, *Applications of a planar separator theorem*, SIAM J. Computing, 9 (1980), 615–627.

[52] M. V. MARATHE, H. BREU, H. B. HUNT, III, S. S. RAVI, AND D. J. ROSENKRANTZ, *Simple heuristics for unit disk graphs*, Networks, 25 (1995), 59–68.

[53] C. NORISHIGE, N. TAKAO, AND S. NOBUJI, *Applications of the Lipton and Tarjan's planar separator theorem*, Journal of Information Processing, 4 (1981), 203–207.

[54] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. II. Algorithmic aspects of tree-width*, Journal of Algorithms, 7 (1986), 309–322.

[55] N. ROBERTSON, P. D. SEYMOUR, AND R. THOMAS, *A survey of linkless embeddings*, in Graph structure theory (Seattle, WA, 1991), Amer. Math. Soc., 1993, 125–136.

[56] P. D. SEYMOUR AND R. THOMAS, *Call routing and the ratcatcher*, Combinatorica, 14 (1994), 217–241.

[57] C. SWAMY AND A. KUMAR, *Primal-dual algorithms for connected facility location problems*, Algorithmica, 40 (2004).

[58] J. WU AND H. LI, *On calculating connected dominating set for efficient routing in ad hoc wireless networks*, in Proc. 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 1999, 7–14.