Bidimensionality and EPTAS

Fedor V. Fomin* Daniel Lokshtanov‡ Venkatesh Raman§ Saket Saurabh§

Abstract

Bidimensionality theory appears to be a powerful framework for the development of meta-algorithmic techniques. It was introduced by Demaine et al. [*J. ACM 2005*] as a tool to obtain sub-exponential time parameterized algorithms for problems on H-minor free graphs. Demaine and Hajiaghayi [*SODA 2005*] extended the theory to obtain polynomial time approximation schemes (PTASs) for bidimensional problems, and subsequently improved these results to EPTASs. Fomin et. al [*SODA 2010*] established a third meta-algorithmic direction for bidimensionality theory by relating it to the existence of linear kernels for parameterized problems. In this paper we revisit bidimensionality theory from the perspective of approximation algorithms and redesign the framework for obtaining EPTASs to be more powerful, easier to apply and easier to understand.

One of the important conditions required in the framework developed by Demaine and Hajiaghayi [*SODA 2005*] is that to obtain an EPTAS for a graph optimization problem Π, we have to know a constant-factor approximation algorithm for Π. Our approach eliminates this strong requirement, which makes it amenable to more problems. At the heart of our framework is a decomposition lemma which states that for "most" bidimensional problems, there is a polynomial time algorithm which given an H-minor-free graph G as input and an $\epsilon > 0$ outputs a vertex set X of size $\epsilon \cdot OPT$ such that the treewidth of $G \setminus X$ is $\mathcal{O}(1/\epsilon)$. Here, OPT is the objective function value of the problem in question This allows us to obtain EPTASs on (apex)-minor-free graphs for all problems covered by the previous framework, as well as for a wide range of packing problems, partial covering problems and problems that are neither closed under taking minors, nor contractions. To the best of our knowledge for many of these problems including Cycle Packing, Vertex-$\mathcal{H}$-Packing, Maximum Leaf Spanning Tree, and Partial r-Dominating Set no EPTASs on planar graphs were previously known.

*Department of Informatics, University of Bergen, Norway. fedor.fomin@ii.uib.no

†Department of Computer Science and Engineering, University of California, San Diego, USA. dlokshtanov@cs.ucsd.edu

§The Institute of Mathematical Sciences, Chennai, India. {vraman|saket}@imsc.res.in

1 Introduction

While most interesting graph problems remain NP complete even when restricted to planar graphs, the restriction of a problem to planar graphs is usually considerably more tractable algorithmically than the problem on general graphs. Over the last four decades, it has been proved that many graph problems on planar graphs admit subexponential time algorithms [18, 25, 33], subexponential time parameterized algorithms [1, 26], (Efficient) Polynomial Time Approximation Schemes ((E)PTAS) [4, 28, 11, 20, 27, 29] and linear kernels [2, 5, 8]. Amazingly, the emerging theory of Bidimensionality developed by Demaine et al. [15, 16, 13] is able to simultaneously explain the tractability of these problems within the paradigms of parameterized algorithms [13], approximation [14] and kernelization [24]. The theory is built on cornerstone theorems from Graph Minors Theory of Robertson and Seymour, and allows not only to explain the tractability of many problems, but also to generalize the results from planar graphs and graphs of bounded genus to graphs excluding a fixed minor. Roughly speaking, a problem is bidimensional if the solution value for the problem on a $k \times k$-grid is $\Omega(k^2)$, and the contraction or removal of an edge does not increase solution value. Many natural problems are bidimensional, including Dominating Set, Feedback Vertex Set, Edge Dominating Set, Vertex Cover, Connected Dominating Set, Cycle Packing, Connected Vertex Cover, and Graph Metric TSP.

A PTAS is an algorithm which takes an instance I of an optimization problem and a parameter $\epsilon > 0$ and, runs in time $n^{\mathcal{O}(f(1/\epsilon))}$, produces a solution that is within a factor ϵ of being optimal. A PTAS with running time $f(1/\epsilon) \cdot n^{\mathcal{O}(1)}$, is called efficient PTAS (EPTAS). Prior to bidimensionality [14], there were two main approaches to design (E)PTASs on planar graphs. The first one was based on the classical Lipton-Tarjan planar separator theorem [32]. The second, more widely used approach was given by Baker [4]. In the Lipton-Tarjan based approach we split the input n-vertex graph into two pieces of approximately equal size using a separator of size $\mathcal{O}(\sqrt{n})$. Then we recursively approximate the problem on the two smaller instances and glue the approximate solutions at the separator. This approach was only applicable to problems where the size of the optimal solutions was at least a constant fraction of n.

The main idea in Baker's approach is to decompose the planar graph into overlapping subgraphs of bounded outerplanarity and then solve the problem optimally in each of these subgraphs using dynamic programming. Later Eppstein [20] generalized this approach to work for larger class of graphs, namely apex minor free graphs. Khanna and Motwani [29] used 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. The same kind of approach is also used by Dawar et al. [11] to obtain EPTASs for every minimization problem definable in first-order logic on every class of graphs excluding a fixed minor. Baker's approach seemed to be limited to "local" graph problems–where one is interested in finding a vertex/edge set satisfying a property that can be checked by looking at constant size neighborhood around each vertex.

Demaine and Hajiaghayi [14] used bidimensionality theory to strengthen and generalize both approaches. In particular they strengthened the Lipton-Tarjan approach significantly by showing that for a magnitude of problems one can find a separator of size $\mathcal{O}(\sqrt{OPT})$ that splits the optimum solution evenly into two pieces. Here OPT is the size of the optimum solution. This allowed them to give EPTASs for several problems on planar graphs and more generally on apex-minor-free graphs or H-minor free graphs. Two important problems to which their approach applies are Feedback vertex Set and Connected Dominating Set. Earlier only a PTAS and no EPTAS for Feedback Vertex Set on planar graphs was known [30]. In addition, they also generalize Baker's approach by allowing more interaction between the overlapping subgraphs.

Comparing the generalized versions of the two approaches, it seems that each has its strengths and weaknesses. In the generalized Lipton-Tarjan approach of Demaine and Hajiaghayi [14] one splits the graph into two pieces recursively. To ensure that the repeated application does not "increase" the approximation factor, in each recursive step one needs to carefully reconstruct the solution from the smaller ones. Additionally, to ensure that the separator splits the optimum solution evenly, the framework of Demaine and Hajiaghayi [14] requires a constant factor approximation for the problem in question. On the other hand, their generalization of Baker's approach essentially identifies a set of vertices or edges that interacts in a limited way with the optimum solution, such that the removal of X from the input graph leaves a graph on which the problem can be solved optimally in polynomial time. The set X could be as large as $\mathcal{O}(n)$ which in some cases makes it difficult to bound the amount of interaction between the set X and the optimum solution.

In this paper we propose a framework which combines the best of both worlds—the generalized Lipton-Tarjan and generalized Baker's approaches. In particular, we show that for most bidimensional problems there is a polynomial time algorithm that given a graph G and an $\epsilon > 0$ outputs a vertex set X of size $\epsilon \cdot OPT$ such that the treewidth of $G \setminus X$ is $\mathcal{O}(1/\epsilon)$. Because the *size* of X is bounded, the interaction between X and the optimum solution is bounded trivially. Since X is only removed once, the difficulty faced by a recursive approach vanishes. In our framework to obtain EPTASs, we demand that the problem in question is "reducible", which is nothing else than that the set X can be removed from the graph, disturbing the optimum solution by at most $\mathcal{O}(\epsilon \cdot OPT)$. Finally, our algorithm to compute X does not require an approximation algorithm for the problem in question, and relies only on a sublinear treewidth bound. For most problems, such a bound can be obtained via bidimensionality, whereas for some problems that are not bidimensional, one can obtain the sublinear treewidth bound directly. This is where our framework differs significantly from the one proposed by Demaine and Hajiaghayi [14]. One of the important condition required to obtain an EPTAS for a graph optimization problem Π using the framework developed in [14] is to have a constant-factor approximation algorithm for Π. Thus our framework removes this stringent condition of demanding approximation algorithm, which makes it amenable to more problems.

Our new framework allows to obtain EPTASs on (apex)-minor-free graphs for all problems covered by the previous framework, as well as for several packing problems, partial covering problems and problems that are neither closed under taking minors nor contractions. For an example consider the Maximum Degree Preserving Spanning Tree problem where given a graph G the objective is to find a spanning tree such that the number of vertices which have the same degree in the tree as in the input graph is maximized. Maximum Degree Preserving Spanning Tree is neither closed under taking minors nor contractions, but one can still apply our framework to obtain an EPTAS for this problem. For another example, consider Cycle Packing, where given a graph G the objective is to find the maximum number of vertex disjoint cycles in G. For this problem, it is not clear how to directly apply the previous framework to obtain an EPTAS. On the other hand, using our framework to obtain an EPTAS for this problem is easy. More generally, we give an EPTAS for the Vertex-$\mathcal{H}$-Packing problem, defined as follows. Let $\mathcal{H}$ be a finite set of connected graphs such that at least one graph in $\mathcal{H}$ is planar. Input to Vertex-$\mathcal{H}$-Packing is a graph G and the objective is to find a maximum size collection of vertex disjoint subgraphs $G_1, \dots, G_k$ of G such that each of them contains some graph from $\mathcal{H}$ as a minor. To the best of our knowledge no EPTASs for Cycle Packing, Vertex-$\mathcal{H}$-Packing, Maximum Leaf Spanning Tree, or Partial r-Dominating Set were previously known, even on planar graphs. Our framework to obtain EPTASs seems to be the most general one could hope for in the context of bidimensionality and approximation.

Our results are less prone to the impracticality issues that follow most results in algorithmic Graph Minors. In particular, it is easy to implement our algorithms in a manner completely independent from results in Graph Minors and only use bounds from graph minor in the analysis. Furthermore, there is evidence [35] that the large hidden constants only arise in the analysis, and that with correct fine tuning algorithms could be made to yield good approximations in practice—at the expense of the rigorous approximation guarantee.

2 Definitions and Notations

In this section we give various definitions which we make use of in the paper. Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. A graph G' is a *subgraph* of G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. The subgraph G' is called an *induced subgraph* of G if $E(G') = \{uv \in E(G) \mid u, v \in V(G')\}$, in this case, G' is also called the subgraph *induced by* V' and denoted by $G[V']$. For a vertex set S, by $G \setminus S$ we denote $G[V(G) \setminus S]$. A graph class $\mathcal{G}$ is *hereditary* if for any graph $G \in \mathcal{G}$ all induced subgraphs of G are in $\mathcal{G}$. By $N(u)$ we denote (open) neighborhood of u, that is, the set of all vertices adjacent to u. Similarly, by $N[u]$ we denote $N(u) \cup \{u\}$. For a subset $D \subseteq V$, we define $N[D] = \cup_{v \in D} N[v]$ and $N(D) = N[D] \setminus D$. The *distance* $d_G(u, v)$ between two vertices u and v of G is the length of the shortest path in G from u to v. Define $B^r_G(v)$ to be the set of vertices within distance at most r from v, including v itself. For a vertex set S, define $B^r_G(v) = \bigcup_{v \in S} B^r_G(v)$. We denote by $\mathbf{tw}(G)$ the *treewidth* of a graph G. (The definition of treewidth can be found in Appendix.)

Minors. Given an edge $e = xy$ of a graph G, the graph G/e is obtained from G by contracting the edge e. That is, the endpoints x and y are replaced by a new vertex v_{xy} which is adjacent to the old neighbors of x and y (except from x and y). A graph H obtained by a sequence of edge-contractions is said to be a *contraction* of G. We denote it by $H \leq_c G$. A graph H is a *minor* of a graph G if H is the contraction of some subgraph of G and we denote it by $H \leq_m G$. We say that a graph G is H*-minor-free* if G does not contain H as a minor. We also say that a graph class $\mathcal{G}_H$ is H*-minor-free* (or, excludes H as a minor) when all its members are H-minor-free. An *apex graph* is a graph obtained from a planar graph G by adding a vertex and making it adjacent to some of the vertices of G. A graph class $\mathcal{G}_H$ is *apex-minor-free* if $\mathcal{G}_H$ excludes a fixed apex graph H as a minor. Let us remark that every planar, and more generally, graph of bounded genus, is an H-minor-free graph for some fixed apex graph H.

Grids and their triangulations. Let r be a positive integer, $r \geq 2$. The $(r \times r)$*-grid* is the Cartesian product of two paths of lengths $r - 1$. A vertex of a grid is a *corner* if it has degree 2. Thus each $(r \times r)$-grid has 4 corners. A vertex of a $(r \times r)$-grid is called *internal* if it has degree 4, otherwise it is called *external*. Let Γ_r be the graph obtained from the $(r \times r)$-grid by triangulating internal faces of the $(r \times r)$-grid such that all internal vertices become of degree 6, all non-corner external vertices are of degree 4, and then one corner of degree two is joined by edges with all vertices of the external face. The graph Γ_6 is shown in Fig. 1.

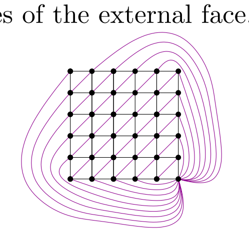

Figure 1: The graph Γ_6.

Counting Monadic Second Order Logic. *Counting monadic second-order logic* (CMSO) is monadic second-order logic (MSO) additionally equipped with an atomic formula $\mathbf{card}_{n,p}(U)$ for testing whether the cardinality of a set U is congruent to n modulo p, where n and p are integers independent of the input graph such that $0 \leq n < p$ and $p \geq 2$. We refer to [3, 9, 10] for a detailed introduction to CMSO. For reader's convenience, we provide the definition of MSO in Appendix. Min-CMSO and Max-CMSO problems are graph optimization problems where the objective is to find a maximum or minimum sized vertex or edge set satisfying a CMSO-expressible property. In particular, in a Min/Max-CMSO graph problem Π we are given a graph G as input. The objective is to find a minimum/maximum cardinality vertex/edge set S such that the CMSO-expressible predicate $P_\Pi(G, S)$ is satisfied.

Bidimensionality and Separability. Our results concern graph optimization problems where the objective is to find a vertex or edge set that satisfies a feasibility constraint and maximizes or minimizes a problemspecific objective function. For a problem Π and vertex (edge) set S let $\phi_\Pi(G,S)$ be the feasibility constraint returning **true** if S is feasible and **false** otherwise. Let $\kappa_\Pi(G,S)$ be the objective function. In most cases, $\kappa_\Pi(G,S)$ will return $|S|$. We will only consider problems where every instance has at least one feasible solution. Let $\mathcal{U}$ be the set of all graphs. For a graph optimization problem Π let $\pi : \mathcal{U} \rightarrow \mathbb{N}$ be a function returning the objective function value of the optimal solution of Π on G. We say that a problem Π is *minor-closed* if $\pi(H) \leq \pi(G)$ whenever H is a minor of G. Similarly, we say Π is *contraction-closed* if $\pi(H) \leq \pi(G)$ whenever H is a contraction of G. We now define bidimensional problems.

Definition 1. ([13, 22]) *A graph optimization problem Π is minor-bidimensional if Π is minor-closed and there is a $\delta > 0$ such that $\pi(R) \geq \delta r^2$ for the $(r \times r)$-grid R. In other words, the value of the solution on R should be at least δr^2. A graph optimization problem Π is called* contraction-bidimensional *if Π is contraction-closed and there is $\delta > 0$ such that $\pi(\Gamma_r) \geq \delta r^2$. In either case, Π is called* bidimensional.

Demaine and Hajiaghayi [14] define the *separation* property for problems, and show how separability together with bidimensionality is useful to obtain EPTASs on H-minor-free graphs. In our setting a slightly weaker notion of separability is sufficient. In particular the following definition is a reformulation of the requirement 3 of the definition of separability in [14] and similar to the definition used in [24] to obtain kernels for bidimensional problems.

Definition 2. *A minor-bidimensional problem Π has the* separation *property if given any graph G and a partition of $V(G)$ into $L \uplus S \uplus R$ such that $N(L) \subseteq S$ and $N(R) \subseteq S$, and given an optimal solution OPT to G, $\pi(G[L]) \leq \kappa_\Pi(G[L], OPT \cap L) + \mathcal{O}(|S|)$ and $\pi(G[R]) \leq \kappa_\Pi(G[R], OPT \cap R) + \mathcal{O}(|S|)$.*

For contraction-bidimensional parameters we have a slightly different definition of the separation property. For a graph G and a partition of $V(G)$ into $L \uplus S \uplus R$ such that $N(L) \subseteq S$ and $N(R) \subseteq S$, we define G_L (G_R) to be the graph obtained from G by contracting every connected component of $G[R]$ ($G[L]$) into the vertex of S with the lowest index.

Definition 3. *A contraction-bidimensional problem has the* separation *property if given any graph G and a partition of $V(G)$ into $L \uplus S \uplus R$ such that $N(L) \subseteq S$ and $N(R) \subseteq S$, and given an optimal solution OPT to G, $\pi(G_L) \leq \kappa_\Pi(G_L, OPT \setminus R) + \mathcal{O}(|S|)$ and $\pi(G_R) \leq \kappa_\Pi(G_R, OPT \setminus L) + \mathcal{O}(|S|)$.*

In Definitions 2 and 3 we slightly misused notation. Specifically, in the case that OPT is an edge set we should not be considering $OPT \setminus R$ and $OPT \setminus L$ but $OPT \setminus E(G[R])$ and $OPT \setminus E(G[L])$ respectively.

Reducibility, η-Transversability and Graph Classes with Truly Sublinear Treewidth. We now define three of the central notions of this article.

Definition 4. *A graph optimization problem Π with objective function κ_Π is called* reducible *if there exists a* Min/Max-CMSO *problem Π' and a function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that*

1. *there is a polynomial time algorithm that given G and $X \subseteq V(G)$ outputs G' such that $\pi'(G') = \pi(G) \pm \mathcal{O}|X|)$ and $\mathbf{tw}(G') \leq f(\mathbf{tw}(G \setminus X))$,*
2. *there is a polynomial time algorithm that given G and $X \subseteq V(G)$, G' and a vertex (edge) set S' such that $P_{\Pi'}(G', S')$ holds, outputs S such that $\phi_\Pi(G,S) =$ **true** and $\kappa_\Pi(G,S) = |S'| \pm \mathcal{O}|X|)$.*

Definition 5. *A graph optimization problem Π is called η-Transversable if there is a polynomial time algorithm that given a graph G outputs a set X of size $\mathcal{O}(\pi(G))$ such that $\mathbf{tw}(G \setminus X) \leq \eta$.*

Definition 6. *A class $\mathcal{G}$ has* truly sublinear treewidth *if there exist constants η, β and λ, such that $\lambda < 1$ and for any graph $G \in \mathcal{G}$ and $X \subseteq V(G)$ such that $\mathbf{tw}(G \setminus X) \leq \eta$, we have $\mathbf{tw}(G) \leq \eta + \beta|X|^\lambda$. We call η, β and λ the* parameters *of $\mathcal{G}$.*

3 Partitioning Graphs of Truly Sublinear Treewidth

We need the following well known lemma, see e.g. [6], on separators in graphs of bounded treewidth.

Lemma 3.1. *Let G be a graph of treewidth at most t and $w : V(G) \rightarrow \mathbb{R}^+ \cup \{0\}$ be a weight function. Then there is a partition of $V(G)$ into $L \uplus S \uplus R$ such that*

- *$|S| \leq t+1$, $N(L) \subseteq S$ and $N(R) \subseteq S$,*
- *every connected component $G[C]$ of $G \setminus S$ has $w(C) \leq w(V)/2$,*
- *$\frac{w(V(G))-w(S)}{3} \leq w(L) \leq \frac{2(w(V(G))-w(S))}{3}$ and $\frac{w(V(G))-w(S)}{3} \leq w(R) \leq \frac{2(w(V(G)-w(S))}{3}$.*

The next lemma is crucial in our analysis.

Lemma 3.2. *Let $\mathcal{G}$ be a hereditary graph class of truly sublinear treewidth with parameters η, λ and β. For any $\epsilon > 1$ there is a γ such that for any $G \in \mathcal{G}$ and $X \subseteq V(G)$ with $\mathbf{tw}(G \setminus X) \leq \eta$ there is $X' \subseteq V(G)$ satisfying $|X'| \leq \epsilon|X|$ and for every connected component C of $G \setminus X'$, we have $|C \cap X| \leq \gamma$ and $|N(C)| \leq \gamma$. Moreover X' can be computed from G and X in polynomial time.*

Proof. For any $\gamma \geq 1$, define $T_\gamma : \mathbb{N} \to \mathbb{N}$ such that $T_\gamma(k)$ is the smallest integer such that if $G \in \mathcal{G}$ and there is a $X \subseteq V(G)$ with $\mathbf{tw}(G \setminus X) \leq \eta$ and $|X| \leq k$, then there is a $X' \subseteq V(G)$ of size at most $T_\gamma(k)$ such that for every connected component C of $G \setminus X'$ we have $|C \cap X| \leq \gamma$ and $|N(C)| \leq \gamma$. Informally, $T_\gamma(k)$ is the minimum size of a vertex set X' such that every connected component C of $G \setminus X'$ has at most γ neighbours in X' and contains at most γ vertices of X, if we know that deleting the k-sized vertex set X from G yields a graph of treewidth η. We will make choices for the constants δ, γ and ρ based on η, λ, β and ϵ. Our aim is to show that $T_\gamma(k) \leq \epsilon k$ for every k.

Observe that for any numbers $a > 0$, $b > 0$, we have $a^\lambda + b^\lambda > (a+b)^\lambda$ since $\lambda < 1$. Thus we have $\rho = \min_{1/3 \leq \alpha \leq 2/3} \alpha^\lambda + (1-\alpha)^\lambda > 1$. We choose $\delta = \frac{(2\epsilon+1)(\beta+\eta+1))}{\rho-1}$ and $\gamma = (\frac{3\delta}{\epsilon})^{\frac{1}{1-\lambda}}$. If $\mathbf{tw}(G \setminus X) \leq \eta$ and $|X| \leq \gamma$ then we set $X' = \emptyset$, so $T_\gamma(k) = 0 \leq \epsilon k$ for $k \leq \gamma$. We now show that if $k \geq \gamma/3$ then $T_\gamma(k) = 0 \leq \epsilon k - \delta k^\lambda$ by induction on k. For the base case if $\gamma/3 \leq k \leq \gamma$ then the choice of γ implies that $\epsilon k - \delta k^\lambda \geq \epsilon \frac{\gamma}{3} - \delta \gamma^\lambda \geq 0 = T_\gamma(k)$.

We now consider $T_\gamma(k)$ for $k > \gamma$. Let $G \in \mathcal{G}$ and $X \subseteq V(G)$ with $\mathbf{tw}(G \setminus X) \leq \eta$ and $|X| \leq k$. The treewidth of G is at most $\eta + \beta k^\lambda$. Construct a weight function $w : V(G) \to \mathbb{N}$ such that $w(v) = 1$, when $v \in X$ and $w(v) = 0$ otherwise. By Lemma 3.1, there is a partition of $V(G)$ into L, S and R such that $|S| \leq \eta + \beta k^\lambda + 1$, $N(L) \subseteq S$, $N(R) \subseteq S$, $|L \cap X| \leq 2k/3$ and $|R \cap X| \leq 2k/3$. Deleting S from the graph G yields two graphs $G[L]$ and $G[R]$ with no edges between them. Thus we put S into X' and then proceed recursively in $G[L \cup S]$ and $G[R \cup S]$ starting from the sets $S \cup (X \cap L)$ and $S \cup (X \cap R)$ in $G[L \cup S]$ and $G[R \cup S]$ respectively. This yields the following recurrence for T_γ.

$$\begin{aligned} T_\gamma(k) \leq \quad & \max_{1/3 \leq \alpha \leq 2/3} T(\alpha k + \eta + \beta k^\lambda + 1) \\ & + \ T((1-\alpha)k + \eta + \beta k^\lambda + 1) \\ & + \ \eta + \beta k^\lambda + 1 \end{aligned}$$

Observe that since $k \geq \gamma$ we have $\alpha k \geq \gamma/3$ and $(1 - \alpha k) \geq \gamma/3$. The induction hypothesis then yields the following inequality.

$$\begin{aligned} T_\gamma(k) \leq \quad & \max_{1/3 \leq \alpha \leq 2/3} T(\alpha k + \eta + \beta k^\lambda + 1) \\ & + \ T((1-\alpha)k + \eta + \beta k^\lambda + 1) \\ & + \ \eta + \beta k^\lambda + 1 \\ \leq \quad & \max_{1/3 \leq \alpha \leq 2/3} \epsilon k - \delta(\alpha k)^\lambda - \delta((1-\alpha)k)^\lambda \\ & + \ (2\epsilon+1)(\beta k^\lambda + \eta + 1) \\ \leq \quad & \max_{1/3 \leq \alpha \leq 2/3} \epsilon k - \delta k^\lambda(\alpha^\lambda + (1-\alpha)^\lambda) \\ & + \ (2\epsilon+1)(\beta k^\lambda + \eta + 1) \\ \leq \quad & \epsilon k - \delta k^\lambda - \delta(\rho - 1)k^\lambda \\ & + \ (2\epsilon+1)(\beta k^\lambda + \eta + 1) \\ \leq \quad & \epsilon k - \delta k^\lambda. \end{aligned}$$

The last inequality holds whenever $\delta(\rho-1)k^\lambda \geq (2\epsilon+1)(\beta k^\lambda + \eta + 1)$, which is ensured by the choice of δ and the fact that $k^\lambda \geq 1$. Thus $T_\gamma(k) \leq \epsilon k$ for all k. Hence there exists a set X' of size at most ϵk such that for every component C of $G \setminus X'$ we have $C \cap X \leq \gamma$ and $|N(C)| \leq \gamma$.

What remains is to show that X' can be computed from G and X in polynomial time. The inductive proof can be converted into a recursive algorithm. The only computationally hard step of the proof is the construction of a tree-decompositon of G in each inductive step. Instead of computing the treewidth exactly we use the $d^* \sqrt{\log \mathbf{tw}(G)}$-approximation algorithm by Feige et al. [21], where d^* is a fixed constant. Thus when we partition $V(G)$ into L, S, and R using Lemma 3.1, the upper bound on the size of S will be $d^*(\eta + \beta k^\lambda)\sqrt{\log(\eta + \beta k^\lambda)}$ instead of $\eta + \beta k^\lambda$. However, for any $\lambda < \lambda' < 1$ there is a β' such that $d^*(\eta + \beta k^\lambda)\sqrt{\log(\eta + \beta k^\lambda)} < \eta + \beta' k^{\lambda'}$. Now we can apply the above analysis with β' instead of β and λ' instead of λ to bound the size of the set X' output by the algorithm. This concludes the proof of the lemma. □

The following corollary is a direct consequence of Lemma 3.2. Nevertheless, we find it worthwhile to mention it separately.

Corollary 3.1. *Let $\mathcal{G}$ be a hereditary graph class of truly sublinear treewidth with parameters η, λ and β. For any $\epsilon > 1$ there is a τ with $\tau = \mathcal{O}(\frac{1}{\epsilon})^{\frac{\lambda}{1-\lambda}})$ such that for any $G \in \mathcal{G}$ and $X \subseteq V(G)$ with $\mathbf{tw}(G \setminus X) \leq \eta$ there is a $X' \subseteq V(G)$ satisfying $|X'| \leq \epsilon|X|$ such that $\mathbf{tw}(G \setminus X') \leq \tau$.*

Proof. We apply Lemma 3.2 on G and X to obtain the set X' of size $\epsilon|X|$. Observe that in the proof of Lemma 3.2, $\gamma = \mathcal{O}(\frac{1}{\epsilon})^{\frac{1}{1-\lambda}})$. The treewidth of $G \setminus X'$ equals the maximum treewidth of a connected component C of $G \setminus X'$. However $|C \cap X| \leq \gamma$ and so $\mathbf{tw}(G[C]) = \mathcal{O}\gamma^\lambda)$, concluding the proof. □

4 Approximation Schemes

Approximation Schemes for η-Transversable problems.

THEOREM 4.1. *Let Π be an η-transversable, reducible graph optimization problem. Then Π has an EPTAS on every graph class $\mathcal{G}$ with truly sublinear treewidth.*

Proof. Let G be the input to Π and $\epsilon > 0$ be fixed. Since Π is η-transversable there is a polynomial time algorithm that outputs a set X such that $|X| \leq \rho_1\pi(G)$ and $\mathbf{tw}(G) \leq \eta$, for a fixed constant ρ_1. Let ϵ' be a constant to be selected later. By Lemma 3.2, there exist γ, $\lambda' < 1$ and β' depending on ϵ', λ, η and β such that given G and X a set X' with the following properties can be found in polynomial time. First $|X'| \leq \epsilon'|X|$, and secondly for every component C of $G \setminus X'$ we have that $C \cap X \leq \gamma$. Thus $\mathbf{tw}(G \setminus X') = \tau \leq \beta'\gamma^{\lambda'} + \eta$. Since Π is reducible there exists a MIN/MAX-CMSO problem Π', a constant ρ_2 and a function $f : \mathbb{N} \to \mathbb{N}$ such that:

1. there is a polynomial time algorithm that given G and $X' \subseteq V(G)$ outputs G' such that $|\pi'(G') - \pi(G)| \leq \rho_2|X'|$ and $\mathbf{tw}(G') \leq f(\tau)$;
2. there is a polynomial time algorithm that given G and $X' \subseteq V(G)$, G' and a vertex (edge) set S' such that $P_{\Pi'}(G', S')$ holds outputs S such that $\phi_\Pi(G, S)$ holds and $|\kappa_\Pi(G, S) - |S'|| \leq \rho_2|X'|$.

We constuct G' from G and X' using the first polynomial time algorithm. Since $\mathbf{tw}(G') \leq f(\tau)$ we can use an extended version of Courcelle's theorem [9, 10] given by Borie et al. [7] to find an optimal solution S' to Π' in $g(\epsilon')|V(G')|$ time. By the properties of the first polynomial time algorithm, $||S'| - \pi(G)| \leq \rho|X'|$ where $\rho = \max(\rho_1, \rho_2)$. We now use the second polynomial time algorithm to construct a solution S to Π from G, X', G' and S'. The properties of the second algorithm ensure $\phi_\Pi(G, S)$ holds and that $|\kappa_\Pi(G, S) - |S'|| \leq \rho|X'|$, and hence $|\kappa_\Pi(G, S) - \pi(G)| \leq 2\rho|X'| \leq 2\rho^2\epsilon'\pi(G)$. Choosing $\epsilon' = \frac{\epsilon}{2\rho^2}$ yields $|\kappa_\Pi(G, S) - \pi(G)| \leq \epsilon\pi(G)$, proving the theorem. $\square$

Approximation Schemes for Bidimensional problems. Now we use Theorem 4.1 to give EPTASs for bidimensional, separable and reducible problems on graphs excluding a fixed H as a minor. In order to do this, we show that H-minor-free graphs have truly sublinear treewidth and that bidimensional and separable problems are η-transversible. To show that H-minor free graphs have truly sublinear treewidth we use the following result.

PROPOSITION 4.1. ([12, 17, 22]) *Let G be a connected graph excluding a fixed graph H as a minor. Then there exists some constant c such that if $\mathbf{tw}(G) \geq c{\cdot}r^2$, then G contains the $r \times r$-grid as a minor. Moreover, if H is an apex graph, then G does not contain Γ_r as a contraction.*

COROLLARY 4.1. *Let $\mathcal{G}_H$ be a class of graphs excluding a fixed graph H as a minor. Then $\mathcal{G}_H$ has truly sublinear treewidth with $\lambda = \frac{1}{2}$.*

Proof. Let ρ be a constant such that any graph $G \in \mathcal{G}_H$ of treewidth at least t contains a $\rho t \times \rho t$ grid as a minor. Let $G \in \mathcal{G}_H$ have a vertex subset X such that $\mathbf{tw}(G \setminus X) \leq \eta$ for a fixed constant η. We prove that $\mathbf{tw}(G) \leq \frac{\eta+1}{\rho}\lceil\sqrt{|X|+1}\rceil$. Suppose not. Then, by Proposition 4.1, G contains a $(\eta+1)\lceil\sqrt{|X|+1}\rceil \times (\eta+1)\lceil\sqrt{|X|+1}\rceil$ grid as a minor. Hence G contains at least $|X|+1$ disjoint $\eta+1 \times \eta+1$ grids as a minor. The set X is disjoint from at least one of these grids, and hence $G \setminus X$ contains a $\eta+1 \times \eta+1$ grid as a minor and has treewidth at least $\eta+1$, yielding the desired contradiction. $\square$

For every fixed integer η we define the η-TRANSVERSAL problem as follows. Input is a graph G, and the objective is to find a minimum cardinality vertex set $S \subseteq V(G)$ such that $\mathbf{tw}(G \setminus S) \leq \eta$. We now give a polynomial time constant factor approximation for the η-TRANSVERSAL problem on H-minor-free graphs.

LEMMA 4.1. *For every integer η and fixed graph H there is a constant c and a polynomial time c-approximation algorithm for the η-TRANSVERSAL problem on H-minor-free graphs.*

Proof. Let X be a smallest vertex set in G such that $\mathbf{tw}(G \setminus X) \leq \eta$. By Lemma 3.2 with $\epsilon = 1/2$ there exists a γ depending only on H and η and a set X' with $|X'| \leq |X|/2$ such that for any component C of $G \setminus X'$, $|C \cap X| \leq \gamma$ and $|N(C)| \leq \gamma$. Since X is the *smallest* set such that $\mathbf{tw}(G \setminus X) \leq \eta$, there is a component C of $G \setminus X'$ with treewidth strictly more than η. Let $Z = N(C)$ and observe that $Z \subseteq X'$ is a set of size at most γ such that C is a conneced component of $G \setminus Z$.

The algorithm proceeds as follows. It tries all possibilities for Z and looks for a connected component C of $G \setminus Z$ such that $\eta < \mathbf{tw}(G[C]) = \mathcal{O}\sqrt{\gamma})$. It solves the η-TRANSVERSAL problem optimally on $G[C]$ by noting that η-TRANSVERSAL can be formulated as a MIN-CMSO problem and applying the algorithm by Borie et al [7]. Let X_C be the solution obtained for $G[C]$. The algorithm adds X_C and $N(C)$ to the solution and repeats this step on $G \setminus (C \cup N(C))$ as long as $\mathbf{tw}(G) \geq \eta$.

Clearly, the set returned by the algorithm is a feasible solution. We now argue that the algorithm is a $(\gamma+1)$-approximation algorithm. Let C_1, C_2, $\ldots C_t$ be the components found by the algorithm in this manner. Since X must contain at least one vertex in each C_i it follows that $t \leq |X|$. Now, for each i, $N(C_i)$ contains at most γ vertices outside of $\bigcup_{j<i} N(C_j)$. Thus $\bigcup_{i \leq t} N(C_i) \leq \gamma|X|$. Furthermore for each C, $|X_C| \leq |X \cap C|$ and hence the size of the returned solution is at most $(\gamma+1)|X|$, which proves the lemma. $\square$

We use Lemma 4.1 in conjunction with the following lemma, which is a combination of Lemmata 3.2 and 3.3 of [24].

Lemma 4.2. ([24]) *Let* Π *be a minor- (contraction-) bidimensional problem with the separation property and* H *be a (apex) graph. There exists a constant* η *such that for every* G *excluding* H *as a minor, there is a subset* $X \subseteq V(G)$ *such that* $|X| = \mathcal{O}(\pi(G))$*, and* $\mathbf{tw}(G \setminus X) \leq \eta$*.*

Together Lemmata 4.1 and 4.2 yield the following corollary.

Corollary 4.2. *Let* Π *be a minor- (contraction-) bidimensional problem with the separation property and* H *be a (apex) graph. There exists a constant* η *such that for every* G *excluding* H *as a minor,* Π *is* η*-transversable on* H*-minor-free graphs.*

Finally, combining Theorem 4.1 with Corollaries 4.1 and 4.2 implies the main theorem of this article.

Theorem 4.2. *Let* Π *be a reducible minor- (contraction-) bidimensional problem with the separation property and* H *be a (apex) graph. There is an EPTAS for* Π *on the class of graphs excluding* H *as a minor.*

5 Applications

5.1 Domination and Connectivity Problems In r-Dominating Set, we are given a graph G, the objective is to find a minimum size subset $S \subseteq V(G)$ such that $B^r_G(S) = V(G)$. For $r = 1$, if we demand that $G[S]$ is connected we obtain the Connected Dominating Set problem. In Connected Vertex Cover we are given a graph G and the objective is to find a minimum size subset $S \subseteq V(G)$ such that $G[S]$ is connected and every edge in $E(G)$ has at least one endpoint in S. It well-known that r-Dominating Set, Connected Dominating Set and Connected Vertex Cover are contraction-bidimensional [13].

Let $V(G) = L \uplus S \uplus R$ with $N(L) \subseteq S$ and $N(R) \subseteq S$. Let G_L and G_R be defined as in Definition 3. Let Z be a minimum size r-dominating set of G and Z_L be a minimum size r-dominating set of G_L. We have that $|Z_L| \leq |Z \setminus R| + |S|$ because $(Z \setminus R) \cup S$ is an r-dominating set of G_L and hence $|Z_L| > |Z \setminus R| + |S|$ contradicts the choice of Z_L. Hence $|Z_L| \leq |Z \setminus R| + \mathcal{O}|S|)$ and r-Dominating Set is separable.

We now show that r-Dominating Set is reducible. Given a graph G and set X, let $G' = G \setminus X$ and let $R = B^r_G(X) \setminus X$. Clearly $\mathbf{tw}(G') = \mathbf{tw}(G \setminus X)$. The annotated problem Π' is to find a minimum sized set $S' \subseteq V(G')$ such that every vertex in $V(G') \setminus (S \cup R)$ is of distance at most r from a vertex in S'. Notice that for any r-dominating set S of G, $S \setminus X$ is a feasible solution to Π' on G'. Conversely, for any feasible solution S' of Π' on G', we have that $S' \cup X$ is an r-dominating set of G. Hence r-Dominating Set is reducible.

The proof that Connected Dominating Set and Connected Vertex Cover are separable are almost identical to the proof for r-Dominating Set with $r = 1$. We only have to note that if Z is an optimal solution to G then $Z \setminus R$ can be made into a feasible solution to G_L by adding S and then observing that $\mathcal{O}(|S|)$ vertices are sufficient to connect the resulting connected components.

We now prove that Connected Dominating Set is reducible. Given a graph G and set X, let $G' = G \setminus X$ and let $R = N(X)$. The annotated problem Π' is to find a minimum sized set $S' \subseteq V(G')$ such that every vertex in $V(G') \setminus (S \cup R)$ has a neighbour in S' and every connected component of $G'[S']$ contains a vertex in R. Notice that for any connected dominating set S of G, $S \setminus X$ is a feasible solution to Π' on G'. Conversely, for any feasible solution S' of Π' on G', we have that $S = S' \cup X$ is a dominating set of G and has at most $|X|$ connected components. Since S is a dominating set it is sufficient to add $2(|X| - 1)$ vertices to S in order to make it a connected dominating set of G. Hence Connected Dominating Set is reducible. The proof that Connected Vertex Cover is reducible is identical.

Finally, let us remark that Connected Vertex Cover is 0-transversible. Given a graph G we find a maximal matching in linear time and output the endpoints of the matching as X. Any vertex cover must contain at least one endpoint from each edge in the matching, and thus $|X| \leq 2\pi(G)$. Also, $\mathbf{tw}(G \setminus X) = 0$.

Lemma 5.1. r-Dominating Set*,* Connected Dominating Set *and* Connected Vertex Cover *are contraction-bidimensional, separable and reducible. Thus they are* η*-transversable on apex-minor-free graphs. Furthermore,* Connected Vertex Cover *is* 0*-transversible on general graphs.*

Max Leaf Spanning Tree. In the Max Leaf Spanning Tree problem we are given a connected graph G and asked to find a spanning tree T of G maximizing the number of leaves of T.

Lemma 5.2. Max Leaf Spanning Tree *is 2-transversible and reducible.*

Proof. We could have shown that the problem is minor-bidimensional and separable, however, just as for Connected Vertex Cover, it is easier to show that the problem is 2-transversible directly. In particular, Kleitman and West [31] have shown that a connected graph which contains no spanning tree with at least k leaves has at most $4k + 2$ vertices of degree at least 3. Thus given a graph we can just return all vertices of degree at least 3, and the remaining graph will have treewidth at most 2. Hence, Max Leaf Spanning Tree is 2-transversible.

We prove that Max Leaf Spanning Tree is reducible. Given a graph G and set X, let $G' = G \setminus X$ and let $R = N(X)$. The annotated problem Π' is to find a maximum size set $S' \subseteq V(G')$ such that every vertex in $S' \setminus R$ has a neighbour outside of S' and every connected component of $G \setminus S$ contains at least one vertex of $R \setminus S$. For a spanning tree T of G let S be the set of leaves of T. Then $S \setminus X$ is a feasible solution to the annotated problem. On the other hand, given a feasible solution S' to Π', observe every component of $G \setminus S'$ contains a vertex of X. We construct a spanning forest F of G with at most $|X|$ components by picking a spanning tree for every component of $G \setminus S$ and for every vertex v in $S \setminus R$ we connect v to a neighbour outside of S. Notice that all vertices of S are leaves of the spanning forest F. From F we can construct a spanning tree T by adding at most $|X| - 1$ edges. Thus T has at least $|S| - 2(|X| - 1)$ leaves and we conclude that Max Leaf Spanning Tree is reducible. □

5.2 Covering and Packing Problems Minor Covering and Packing. We give below a few generic problems each of which subsumes many problems in itself and fit in our framework. Let $\mathcal{H}$ be a finite set of connected graphs such that at least one graph in $\mathcal{H}$ is planar.

Vertex-$\mathcal{H}$-Covering
Input: A graph G
Objective: Find a minimum size set $S \subseteq V(G)$ such that $G \setminus S$ does not contain any of the graphs from $\mathcal{H}$ as a minor.

Vertex-$\mathcal{H}$-Packing
Input: A graph G.
Objective: Find a maximum size collection of vertex disjoint subgraphs $G_1, \ldots, G_k$ of G such that each of them contains some graph from $\mathcal{H}$ as a minor.

It it easy to see that both Vertex-$\mathcal{H}$-Covering and Vertex-$\mathcal{H}$-Packing are minor-closed problems. Now, let h be the size of the smallest planar graph H in $\mathcal{H}$. By a result of Robertson et al. [34], H is a minor of the $(t \times t)$-grid, where $t = 14|V(H)| - 24$. Consider a $(r \times r)$-grid F. F contains r^2/t^2 disjoint H-minors. Any covering of F must pick at least one vertex from each of these minors, therefore both Vertex-$\mathcal{H}$-Covering and Vertex-$\mathcal{H}$-Packing are minor-bidimensional.

We now prove that Vertex-$\mathcal{H}$-Covering is separable. Given a graph G and a partition of $V(G)$ into L, S and R such that $N(L) \subseteq S$ and $N(R) \subseteq S$, let Z be a set of minimum size such that $G \setminus Z$ contains no graph of $\mathcal{H}$ as a minor. Consider the smallest set Z_L such that $G[L] \setminus Z_L$ contains no graph of $\mathcal{H}$ as a minor. If $|Z \cap L| < |Z_L|$, this contradicts the choice of Z_L since $G[L] \setminus (Z \cap L)$ does not contain a graph of $\mathcal{H}$ as a minor. The proof for Z_R is identical, thus Vertex-$\mathcal{H}$-Covering is separable.

The proof of separability of Vertex-$\mathcal{H}$-Packing goes along the same lines as the proof for Vertex-$\mathcal{H}$-Covering, but with a few notable differences. In particular, we formalize Vertex-$\mathcal{H}$-Packing as a graph optimization problem where we seek an edge set $Z \subseteq E(G)$. The objective function, κ_{COV}, *counts* the number of connected components of the subgraph formed by Z that contain at least one copy of a graph in $\mathcal{H}$ as a minor, and all edge subsets are considered feasible.

Given a graph G and a partition of $V(G)$ into L, S and R such that $N(L) \subseteq S$ and $N(R) \subseteq S$, let Z be an edge set maximizing $\kappa_{COV}(G, Z)$ and Z_L be an edge set maximizing $\kappa_{COV}(G[L], Z_L)$. By the choice of Z_L we have $\kappa_{COV}(G[L], Z_L) \geq \kappa_{COV}(G[L], Z \cup E(G[L]))$. The proof for Z_R is identical, hence Vertex-$\mathcal{H}$-Packing is separable.

Finally, it is easy to see that both Vertex-$\mathcal{H}$-Covering and Vertex-$\mathcal{H}$-Packing are reducible. Given G and X we let $G' = G \setminus X$. For Vertex-$\mathcal{H}$-Covering X can be added to the an optimal solution in G' at the cost of $|X|$. For Vertex-$\mathcal{H}$-Packing at most $|X|$ of the minors of graphs in $\mathcal{H}$ contained a vertex in X and got removed when X was deleted. Expressing both problems as Min/Max-CMSO problems is routine.

Lemma 5.3. Vertex-$\mathcal{H}$-Covering *and* Vertex-$\mathcal{H}$-Packing *are minor-bidimensional, separable and reducible.*

VERTEX-$\mathcal{H}$-COVERING contains various problems as a special case, for example: (a) VERTEX COVER by letting $\mathcal{H}$ contain a single graph on a single edge, (b) FEEDBACK VERTEX SET by setting $\mathcal{H}$ to contain a single graph, the complete graph on 3 vertices K_3; (c) DIAMOND HITTING SET by letting $\mathcal{H}$ contain a single graph, the comlete graph on 4 vertices K_4 minus a single edge. Other choices for $\mathcal{H}$ lead to vertex deletion into outerplanar graphs, series-parallel graphs, graphs of constant treewidth (η-TRANSVERSAL) or pathwidth. On the other hand, VERTEX-$\mathcal{H}$-PACKING contains problems like CYCLE PACKING as a special case.

Subgraph Covering and Packing. Now we consider problems about covering and packing subgraphs. These problems can be handled in much the same way as covering and packing minors. Let $\mathcal{S}$ be a finite set of connected graphs.

VERTEX-$\mathcal{S}$-COVERING *Input:* A graph G *Objective:* Find a minimum size set $S \subseteq V(G)$ such that $G \setminus S$ does not contain any of the graphs from $\mathcal{S}$ as a subgraph.
VERTEX-$\mathcal{S}$-PACKING *Input:* A graph G. *Objective:* Find a maximum size collection of vertex disjoint subgraphs $G_1, \ldots, G_k$ of G such that each of them contains some graph from $\mathcal{S}$ as a subgraph.

LEMMA 5.4. VERTEX-$\mathcal{S}$-COVERING *or* VERTEX-$\mathcal{S}$-PACKING *pre-processed with the Redundant Vertex Rule are η-transversible and reducible.*

Proof. We will not show that VERTEX-$\mathcal{S}$-COVERING or VERTEX-$\mathcal{S}$-PACKING are bidimensional. Instead, we will give a reduction rule, and show that instances reduced according to this rule have an r-dominating set of size $\mathcal{O}(OPT)$, where r is the maximum size of a graph in $\mathcal{S}$. Since r-DOMINATING SET is η-transversible there is an algorithm that in polynomial time outputs a set $X \subseteq V(G)$ of size $\mathcal{O}(OPT)$ such that $\mathbf{tw}(G \setminus X) \leq \eta$. Hence the pre-processed version of VERTEX-$\mathcal{S}$-COVERING and VERTEX-$\mathcal{S}$-PACKING is η-transversible.

Consider the following rule, the *Redundant Vertex Rule*. Given as input G to VERTEX-$\mathcal{S}$-COVERING or VERTEX-$\mathcal{S}$-PACKING remove all vertices that are not part of any subgraph isomorphic to any graph in $\mathcal{S}$. We can perform the Redundant Vertex Rule in $\mathcal{O}(|V| \cdot |\mathcal{S}|)$ time by looking at a small ball around every vertex v and check whether the ball contains a subgraph isomorphic to a graph in $\mathcal{S}$ that contains v. This algorithm to check a subgraph isomorphic to a given graph containing a particular vertex appears in an article of Eppstein [19].

Consider an instance G of VERTEX-$\mathcal{S}$-COVERING reduced according to the Redundant Vertex Rule, and let S be an optimal solution to G. Since X hits all copies of graphs in $\mathcal{S}$ occuring in G and every vertex in G appears in some copy of a graph in $\mathcal{S}$ it follows that X is a r-dominating set of G, where r is the maximum size of a graph in $\mathcal{S}$. Finally, consider an instance G of VERTEX-$\mathcal{S}$-PACKING reduced according to the Redundant Vertex Rule, and consider an optimal solution $G_1, \ldots, G_{OPT}$ such that for every i, G_i contains a graph. $\square$

5.3 Partial Domination and Covering In the PARTIAL r-DOMINATING SET problem we are given a graph G together with an integer $t \leq |V(G)|$. The objective is to find a minimum size set S such that $|B^r_G(S)| \geq t$. In PARTIAL VERTEX COVER we are given a graph G together with an integer $t \leq |E(G)|$ and the objective is to find a minimum size vertex set S such that $|\{uv \in E : u \in S \vee v \in S\}| \geq t$. We will call $\{uv \in E : u \in S \vee v \in S\}$ the set of edges *covered* by S. PTAS for PARTIAL VERTEX COVER on planar graphs was given in [27]. We are not aware of any PTAS for PARTIAL r-DOMINATING SET.

We will not show that PARTIAL r-DOMINATING SET and PARTIAL VERTEX COVER are bidimensional, instead we will directly construct a EPTASs for these problems on apex-minor-free graphs using the tools developed so far. We will use OPT for the size of an optimal solution to our instances. Let H be a fixed apex graph, our input graphs will exclude H as a minor. We employ an algorithm of Fomin et al. [23]. They give an algorithm for solving PARTIAL r-DOMINATING SET in time $2^{\mathcal{O}(r\sqrt{OPT})}n^{\mathcal{O}1)}$ and PARTIAL VERTEX COVER in time $2^{\mathcal{O}(\sqrt{OPT})}n^{\mathcal{O}1)}$. A key part of their algorithm for PARTIAL r-DOMINATING SET is a polynomial time algorithm ([23], Lemma 5) that given a graph G together with integers t and k returns an induced subgraph G' of G such that G has a k-sized vertex set S such that $|B^r_G(S)| \geq t$ if and only if G' has a k-sized vertex set S' such that $|B^r_{G'}(S')| \geq t$. Furthermore, G' has a $3r$-dominating set of size k. Our EPTAS loops over all possible values of k and for each such value produces G'_k from G, t and k using Lemma 5 of [23]. If G'_k has less than t vertices, then G'_k cannot have any set which covers at least t vertices, and so, neither can G. If G'_k has at least t vertices, we proceed with the following subroutine.

Just as in the proof of Theorem 4.1, let ϵ' be a constant to be chosen later. By construction G'_k has

$3r$-dominating set of size k. Since $3r$-DOMINATING SET is η-transversible there is a polynomial time algorithm that outputs a set X of size at most ρk such that $\mathbf{tw}(G'_k \setminus X) \leq \eta$. By Lemma 3.2 there is a polynomial time algorithm that computes a set X' of size at most $\epsilon' \rho k$ such that $\mathbf{tw}(G'_k \setminus X') \leq \delta$ for a constant δ depending only on η and H. We put all vertices in X' in our solution. Specifically, we remove X' from G'_k and put all other vertices of $B^r_{G'_k}(X')$ into a set R. Using standard dynamic programming (or by formulating the problem in an extended version of MSO [3]) on graphs of bounded treewidth, we can find a minimum size set $S' \subseteq V(G'_k) \setminus X'$ such that $|X'| + |R \cup B^r_{G'_k \setminus X'}(S')| \geq t$ in time $f(\delta)n^{\mathcal{O}1)}$. The subroutine returns the set $S' \cup X'$ as a solution.

Since G' is an induced subgraph of G, any solution $S = S' \cup X'$ returned by the subroutine covers at least t vertices in G. We return the smallest S as our $(1+\epsilon)$-approximate solution. In the iteration of the outer loop where $k = OPT$ we have that G'_k has a set Z of size OPT that covers t vertices in G'. Observe that $Z \setminus X'$ covers at least $t - |B^r_{G'_k}(X')|$ of $V(G'_k) \setminus B^r_{G'_k}(X')$ in the graph $G'_k \setminus X'$. Thus the solution returned by the dynamic programming algorithm has size at most $|Z \setminus X'| \leq |Z| = OPT$ and the solution returned by the subroutine in this iteration is at most $OPT + |X'| \leq OPT(1+\epsilon'\rho)$. Chosing $\epsilon' = \epsilon/\rho$ concludes the analysis of our EPTAS for PARTIAL r-DOMINATING SET. An EPTAS for PARTIAL VERTEX COVER can be constructed in a similar manner.

LEMMA 5.5. *There is an EPTAS for* PARTIAL r-DOMINATING SET *and* PARTIAL VERTEX COVER *on apex-minor-free graphs.*

Finally by applying Theorems 4.1 and 4.2 together with Lemmata 5.1, 5.2, 5.3, 5.4 and 5.5 we get the following corollary.

COROLLARY 5.1. FEEDBACK VERTEX SET*,* VERTEX COVER*,* CONNECTED VERTEX COVER*,* CYCLE PACKING*,* DIAMOND HITTING SET*,* MINIMUM-VERTEX FEEDBACK EDGE SET*,* VERTEX-$\mathcal{H}$-PACKING*,* VERTEX-$\mathcal{H}$-COVERING*,* MAXIMUM INDUCED FOREST*,* MAXIMUM INDUCED BIPARTITE SUBGRAPH *and* MAXIMUM INDUCED PLANAR SUBGRAPH *admit an EPTAS on H-minor-free graphs.* EDGE DOMINATING SET*,* DOMINATING SET*,* r-DOMINATING SET*,* q-THRESHOLD DOMINATING SET*,* CONNECTED DOMINATING SET*,* DIRECTED DOMINATION*,* r-SCATTERED SET*,* MINIMUM MAXIMAL MATCHING*,* INDEPENDENT SET*,* MAXIMUM FULL-DEGREE SPANNING TREE*,* MAX INDUCED AT MOST d-DEGREE SUBGRAPH*,* MAX INTERNAL SPANNING TREE*,* INDUCED MATCHING*,* TRIANGLE PACKING*,* VERTEX-$\mathcal{S}$-COVERING*,* VERTEX-$\mathcal{S}$-PACKING PARTIAL r-DOMINATING SET *and* PARTIAL VERTEX COVER *admit an EPTAS on apex-minor-free graphs.*

It should be noted that for a fixed ϵ, the treewidth τ in Corollary 3.1 is $\mathcal{O}(1/\epsilon)$. For H-minor-free graphs one can compute the set X' from G and X using the procedure described in the last paragraph of the proof of Lemma 3.2 but using the constant factor approximation for treewidth on H-minor-free graphs [21] instead of the approximation algorithm for general graphs. For many problems discussed in this paper, the MSO-based algorithms on graphs of bounded treewidth could be replaced by standard dynamic programming algorithms with running time $2^{\mathcal{O}\mathbf{tw}(G))}n$ or $2^{\mathcal{O}\mathbf{tw}(G)\log(\mathbf{tw}(G)))}n$. On H-minor-free graphs this leads to EPTASs with running times on the form $2^{\mathcal{O}1/\epsilon)}n + n^{\mathcal{O}1)}$ or $2^{\mathcal{O}1/\epsilon\log(1/\epsilon))}n + n^{\mathcal{O}1)}$, which is comparable to the fastest previously known results.

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), pp. 461–493.

[2] J. ALBER, M. R. FELLOWS, AND R. NIEDERMEIER, *Polynomial-time data reduction for dominating set*, Journal of the ACM, 51 (2004), pp. 363–384.

[3] S. ARNBORG, J. LAGERGREN, AND D. SEESE, *Easy problems for tree-decomposable graphs*, J. Algorithms, 12 (1991), pp. 308–340.

[4] B. S. BAKER, *Approximation algorithms for NP-complete problems on planar graphs*, J. ACM, 41 (1994), pp. 153–180.

[5] H. BODLAENDER, F. V. FOMIN, D. LOKSHTANOV, E. PENNINKX, S. SAURABH, AND D. M. THILIKOS, *(Meta) Kernelization*, in FOCS 2009, IEEE, 2009, pp. 629–638.

[6] H. L. BODLAENDER, *A partial k-arboretum of graphs with bounded treewidth*, Theor. Comp. Sc., 209 (1998), pp. 1–45.

[7] 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), pp. 555–581.

[8] J. CHEN, H. FERNAU, I. A. KANJ, AND G. XIA, *Parametric duality and kernelization: Lower bounds and upper bounds on kernel size*, SIAM J. Comput., 37 (2007), pp. 1077–1106.

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

[10] ———, *The expression of graph properties and graph transformations in monadic second-order logic*, Handbook of Graph Grammars, (1997), pp. 313–400.
[11] A. Dawar, M. Grohe, S. Kreutzer, and N. Schweikardt, *Approximation schemes for first-order definable optimisation problems*, LICS 2006, (2006), pp. 411–420.
[12] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos, *Bidimensional parameters and local treewidth*, SIAM J. Discrete Math., 18 (2004/05), pp. 501–511.
[13] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos, *Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs*, J. ACM, 52 (2005), pp. 866–893.
[14] E. D. Demaine and M. Hajiaghayi, *Bidimensionality: New connections between FPT algorithms and PTASs*, in SODA 2005, ACM-SIAM, 2005, pp. 590–601.
[15] ———, *Bidimensionality*, in Encyclopedia of Algorithms, M.-Y. Kao, ed., Springer, 2008.
[16] ———, *The bidimensionality theory and its algorithmic applications*, Comput. J., 51 (2008), pp. 292–302.
[17] E. D. Demaine and M. Hajiaghayi, *Linearity of grid minors in treewidth with applications through bidimensionality*, Combinatorica, 28 (2008), pp. 19–36.
[18] F. Dorn, F. V. Fomin, and D. M. Thilikos, *Subexponential parameterized algorithms*, Computer Science Review, 2 (2008), pp. 29–39.
[19] D. Eppstein, *Subgraph isomorphism in planar graphs and related problems*, J. Graph Algorithms and Applications, 3 (1999), pp. 1–27.
[20] ———, *Diameter and treewidth in minor-closed graph families*, Algorithmica, 27 (2000), pp. 275–291.
[21] U. Feige, M. Hajiaghayi, and J. R. Lee, *Improved approximation algorithms for minimum-weight vertex separators*, in Proceedings of the 37th annual ACM Symposium on Theory of computing (STOC 2005), New York, 2005, ACM Press, pp. 563–572.
[22] F. V. Fomin, P. A. Golovach, and D. M. Thilikos, *Contraction bidimensionality: the accurate picture*, in ESA 2009, vol. 5757 of LNCS, Sptinger, 2009, pp. 706–717.
[23] F. V. Fomin, D. Lokshtanov, V. Raman, and S. Saurabh, *Subexponential algorithms for partial cover problems*, in FSTTCS, 2009, pp. 193–201.
[24] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos, *Bidimensionality and kernels*, in SODA 2010, ACM-SIAM, 2010, pp. 503–510.
[25] F. V. Fomin and D. M. Thilikos, *New upper bounds on the decomposability of planar graphs*, Journal of Graph Theory, 51 (2006), pp. 53–81.
[26] F. V. Fomin and D. M. Thilikos, *New upper bounds on the decomposability of planar graphs*, J. Graph Theory, 51 (2006), pp. 53–81.
[27] R. Gandhi, S. Khuller, and A. Srinivasan, *Approximation algorithms for partial covering problems*, J. Algorithms, 53 (2004), pp. 55 – 84.
[28] M. Grohe, *Local tree-width, excluded minors, and approximation algorithms*, Combinatorica, 23 (2003), pp. 613–632.
[29] S. Khanna and R. Motwani, *Towards a syntactic characterization of ptas*, in STOC 1996, ACM, 1996, pp. 329–337.
[30] J. Kleinberg and A. Kumar, *Wavelength conversion in optical networks*, Journal of Algorithms, 38 (2001), pp. 25–50.
[31] D. J. Kleitman and D. B. West, *Spanning trees with many leaves*, SIAM J. Discrete Math., 4 (1991), pp. 99–106.
[32] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math., 36 (1979), pp. 177–189.
[33] ———, *Applications of a planar separator theorem*, SIAM J. Comput., 9 (1980), pp. 615–627.
[34] N. Robertson, P. D. Seymour, and R. Thomas, *Quickly excluding a planar graph*, J. Comb. Theory Series B, 62 (1994), pp. 323–348.
[35] S. Tazari and M. Müller-Hannemann, *Dealing with large hidden constants: Engineering a planar steiner tree ptas*, in ALENEX, SIAM, 2009, pp. 120–131.

A Treewidth.

A *tree decomposition* of a graph G is a pair $(\mathcal{X}, T)$, where T is a tree and $\mathcal{X} = \{X_i \mid i \in V(T)\}$ is a collection of subsets of V such that the following conditions are satisfied.

1. $\bigcup_{i \in V(T)} X_i = V(G)$.
2. For each edge $xy \in (G)$, $\{x, y\} \subseteq X_i$ for some $i \in V(T)$.
3. For each $x \in V(G)$ the set $\{i \mid x \in X_i\}$ induces a connected subtree of T.

The *width* of the tree decomposition is $\max_{i \in V(T)} |X_i| - 1$. The *treewidth* of a graph G, $\mathbf{tw}(G)$, is the minimum width over all tree decompositions of G.

B MSO

The syntax of MSO of graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, $\Leftrightarrow$, $\Rightarrow$, variables for vertices, edges, set of vertices and set of edges, the quantifiers $\forall$, $\exists$ that can be applied to these variables, and the following five binary relations:

1. $u \in U$ where u is a vertex variable and U is a vertex set variable.
2. $d \in D$ where d is an edge variable and D is an edge set variable.
3. $\mathbf{inc}(d, u)$, where d is an edge variable, u is a vertex variable, and the interpretation is that the edge d is incident on the vertex u.

4. $\mathbf{adj}(u, v)$, where u and v are vertex variables u, and the interpretation is that u and v are adjacent.
5. Equality of variables, $=$, representing vertices, edges, set of vertices and set of edges.