FPT Approximations for Packing and Covering Problems Parameterized by Elimination Distance and Even Less

Tanmay Inamdar
University of Bergen, Norway

Lawqueen Kanesh
Indian Institute of Technology Jodhpur, India

Madhumita Kundu
University of Bergen, Norway

M. S. Ramanujan
University of Warwick, Coventry, UK

Saket Saurabh
Institute of Mathematical Sciences, Chennai, India
University of Bergen, Norway

Abstract

For numerous graph problems in the realm of parameterized algorithms, using the size of a smallest deletion set (called a modulator) into well-understood graph families as parameterization has led to a long and successful line of research. Recently, however, there has been an extensive study of structural parameters that are potentially much smaller than the modulator size. In particular, recent papers [Jansen et al. STOC 2021; Agrawal et al. SODA 2022] have studied parameterization by the size of the modulator to a graph family $\mathcal{H}$ ($\mathbf{mod}_{\mathcal{H}}(\cdot)$), elimination distance to $\mathcal{H}$ ($\mathbf{ed}_{\mathcal{H}}(\cdot)$), and $\mathcal{H}$-treewidth ($\mathbf{tw}_{\mathcal{H}}(\cdot)$). These parameters are related by the fact that $\mathbf{tw}_{\mathcal{H}}$ lower bounds $\mathbf{ed}_{\mathcal{H}}$, which in turn lower bounds $\mathbf{mod}_{\mathcal{H}}$. While these new parameters have been successfully exploited to design fast exact algorithms their utility (especially that of $\mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$) in the context of approximation algorithms is mostly unexplored.

The conceptual contribution of this paper is to present novel algorithmic meta-theorems that expand the impact of these structural parameters to the area of FPT Approximation, mirroring their utility in the design of exact FPT algorithms. Precisely, we show that if a covering or packing problem is definable in Monadic Second Order Logic and has a property called Finite Integer Index (FII), then the existence of an FPT Approximation Scheme (FPT-AS, i.e., $(1 \pm \epsilon)$-approximation) parameterized by $\mathbf{mod}_{\mathcal{H}}(\cdot), \mathbf{ed}_{\mathcal{H}}(\cdot)$, and $\mathbf{tw}_{\mathcal{H}}(\cdot)$ is in fact equivalent. As a consequence, we obtain FPT-ASes for a wide range of covering, packing, and domination problems on graphs with respect to these parameters. In the process, we show that several graph problems, that are W[1]-hard parameterized by $\mathbf{mod}_{\mathcal{H}}$, admit FPT-ASes not only when parameterized by $\mathbf{mod}_{\mathcal{H}}$, but even when parameterized by the potentially much smaller parameter $\mathbf{tw}_{\mathcal{H}}(\cdot)$. In the spirit of [Agrawal et al. SODA 2022], our algorithmic results highlight a broader connection between these parameters in the world of approximation. As concrete exemplifications of our meta-theorems, we obtain FPT-ASes for well-studied graph problems such as Vertex Cover, Feedback Vertex Set, Cycle Packing and Dominating Set, parameterized by $\mathbf{tw}_{\mathcal{H}}(\cdot)$ (and hence, also by $\mathbf{mod}_{\mathcal{H}}(\cdot)$ or $\mathbf{ed}_{\mathcal{H}}(\cdot)$), where $\mathcal{H}$ is any family of minor free graphs.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Approximation algorithms analysis

Keywords and phrases FPT-AS, $\mathcal{F}$-Deletion, Packing, Elimination Distance, Elimination Treewidth

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2023.28

Related Version *Full Version*: https://arxiv.org/abs/2310.03469

43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023).
Editors: Patricia Bouyer and Srikanth Srinivasan; Article No. 28; pp. 28:1–28:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding *Tanmay Inamdar*: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416).
M. S. Ramanujan: Supported by Engineering and Physical Sciences Research Council (EPSRC) grants EP/V007793/1 and EP/V044621/1.
Saket Saurabh: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416), and Swarnajayanti Fellowship (No. DST/SJF/MSA01/2017-18).

1 Introduction

One of the most widely studied graph problems in the area of parameterized complexity is the $\mathcal{F}$-Vertex Deletion problem, where the input is a graph G and a number k and the goal is – "Compute a set of at most k vertices whose deletion places the resulting graph in the graph family $\mathcal{F}$ or correctly conclude that such a set does not exist." A solution to an instance of $\mathcal{F}$-Vertex Deletion is called a *modulator* into $\mathcal{F}$ and there are numerous results in parameterized complexity on exploiting modulators into various graph families to design algorithms. Much of this research has been motivated by the fact that inputs that have modulators of small size into $\mathcal{F}$ turn out to be tractable for many problems that are NP-complete in general while being polynomial-time solvable on $\mathcal{F}$. In other words, it is possible to take efficient algorithms for some problems on graphs in $\mathcal{F}$, and lift them to efficient algorithms for these problems on graphs that are not necessarily in $\mathcal{F}$, but have a small vertex modulator into $\mathcal{F}$, i.e., graphs that are "close" to $\mathcal{F}$. This leads to fixed-parameter algorithms for these problems (i.e., running time bounded by $f(k) \cdot n^{O(1)}$, where k is the modulator size and n is the input size) under these parameterizations. Using the size of the smallest vertex modulator of a graph into tractable graph families or "the distance from triviality" methodology [12] has therefore become a rich source of interesting and useful parameters for graph problems over the last two decades.

In light of the success of this line of research, recent years have seen a shift towards identifying and exploring the power of "hybrid" parameters that are upper bounded by the modulator size as well as certain graph-width measures and can be arbitrarily (and simultaneously) smaller than both the modulator size and these graph-width measures. Two specific parameters studied in this line of research are: $\mathcal{H}$-elimination distance and $\mathcal{H}$-treewidth of G. The $\mathcal{H}$-elimination distance of a graph G (denoted $\mathbf{ed}_{\mathcal{H}}(G)$) was introduced by Bulian and Dawar [2] and roughly speaking, it expresses the number of rounds needed to obtain a graph in $\mathcal{H}$ by removing one vertex from every connected component in each round. We refer the reader to Section 2 for a more formal definition. Note that $\mathbf{ed}_{\mathcal{H}}(G)$ (respectively, $\mathbf{tw}_{\mathcal{H}}(G)$) can be arbitrarily smaller than both $\mathbf{mod}_{\mathcal{H}}(G)$ and the treedepth of G (respectively, the treewidth of G). For example, let $\mathcal{H}$ be the (infinite) family of complete graphs, and let G be a graph that contains a vertex v that is adjacent to some (non-empty) subset of vertices from a clique of size t, for some $t \geq 1$. Then, $\mathbf{tw}(G) = t - 1$, whereas $\mathbf{tw}_{\mathcal{H}}(G) = 0$ (in fact, even $\mathbf{ed}_{\mathcal{H}}(G) = 0$). Further, $\mathbf{tw}_{\mathcal{H}}(G)$ itself can be arbitrarily smaller than $\mathbf{ed}_{\mathcal{H}}(G)$ (see [9, 15] for some examples).

Recent work by Agrawal et al. [1] and Jansen et al. [15] show that for many basic graph problems in the literature and well-understood graph families $\mathcal{F}$, one can indeed obtain FPT algorithms parameterized by $\mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$, thus expanding the notion of useful distance from triviality to encompass these parameters as well. Agrawal et al. [1] also showed a tight connection between $\mathbf{mod}_{\mathcal{H}}, \mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$ by showing that for these problems, having an FPT algorithm parameterized by $\mathbf{mod}_{\mathcal{H}}$ was sufficient to obtain FPT algorithms parameterized by the other two "smaller" parameters. In addition to these results, there is also some recent work on computing $\mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$, where $\mathcal{H}$ is the family of bipartite graphs [14], or a minor-closed family [19].

Despite these leaps in our understanding of the parameterized complexity of many problems, limitations remain. For instance, by requiring that the problem be FPT parameterized by $\mathbf{mod}_{\mathcal{H}}$, we are implicitly requiring that the problem be polynomial-time solvable on the class $\mathcal{H}$. This rules out meaningful results for many basic problems and established graph families $\mathcal{H}$. For instance, it is not interesting to study Vertex Cover parameterized by $\mathbf{mod}_{\mathcal{H}}$ when $\mathcal{H}$ is the class of planar graphs since Vertex Cover is NP-complete on planar graphs [10]. However, it is efficiently approximable on planar graphs (i.e., even has an Efficient PTAS) [7]. This state of the art brings us to the following two natural questions and is the main motivation behind this work.

Question 1: Can good *approximation algorithms* for a problem on the class $\mathcal{H}$ be used to obtain good FPT approximation algorithms for the same problem parameterized by $\mathbf{mod}_{\mathcal{H}}$?

Question 2: Could one obtain positive answers to the above question, but for the parameters $\mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$?

The notion of an FPT approximation algorithm is easily motivated by the simultaneous existence of fixed-parameter intractability results as well as polynomial-time inapproximability results for numerous problems in the literature. Hence, the topic of FPT approximation, has been an extremely active area of research in the last decade. For a comprehensive survey of the state of the art, we refer the reader to the survey by Feldman et al. [5].

We note that we are not the first to study Question 1. In Marx's classic survey [18] on parameterized approximation, he outlines an FPT approximation algorithm for Chromatic Number parameterized by the modulator to planar graphs. More recently, Demaine et al. [4] studied this question systematically, albeit restricted to polynomial-time approximation. They developed a general theorem that gives sufficient conditions on the problem in order to guarantee an affirmative answer to this question when the class $\mathcal{H}$ has bounded treewidth or arboricity, but in their context, the approximation ratio of the algorithms depends on $\mathbf{mod}_{\mathcal{H}}$. However, Question 1 in the setting of FPT approximation is largely unexplored. To the best of our knowledge, Question 2 has not been considered in the literature and this is the main focus of our paper.

1.1 Our contributions

The main conceptual message of the paper is the following meta-result (stated informally):

If a problem can be captured by a very expressible logic fragment (i.e, Counting Monadic Second Order Logic (CMSO)) and has an appropriate graph replacement subroutine (i.e., has Finite Integer Index (FII)) and fulfills a few other mild requirements, then the existence of an FPT-AS for the problem parameterized by any of the three parameters $\mathbf{mod}_{\mathcal{H}}, \mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$ is equivalent to each other.

The formal version of our first meta-theorem in the context of the well-studied family of vertex-deletion problems is given below. Say that a family of graphs is *well-behaved* if it is hereditary and closed under disjoint union.

▶ **Theorem 1.** *Let $\mathcal{H}, \mathcal{F}$ be well-behaved families of graphs, where $\mathcal{F}$ is CMSO-definable. Suppose $\Pi =$ VERTEX DELETION TO $\mathcal{F}$ has* FII*. Then, the following statements are equivalent.*

1. Π *admits an* FPT-AS *parameterized by* $\mathbf{mod}_{\mathcal{H}}(\cdot)$ *and* ϵ*.*
2. Π *admits an* FPT-AS *parameterized by* $\mathbf{ed}_{\mathcal{H}}(\cdot)$ *and* ϵ*.*
3. Π *admits an* FPT-AS *parameterized by* $\mathbf{tw}_{\mathcal{H}}(\cdot)$ *and* ϵ*.*

In the full version of the paper, we also prove meta-theorems similar to Theorem 1 where Π is $\mathcal{F}$-SUBGRAPH PACKING (i.e., pack maximum number of vertex-disjoint subgraphs isomorphic to graphs in $\mathcal{F}$) or $\mathcal{F}$-MINOR PACKING (i.e., pack maximum number of vertex-disjoint minor-models of graphs in $\mathcal{F}$).

In order to invoke these equivalence theorems, we first show that a wide range of graph problems, that are W[1]-hard (or para-NP hard) parameterized by $\mathbf{mod}_{\mathcal{H}}$, admit FPT Approximation Schemes (FPT-ASes, i.e., $(1 \pm \epsilon)$-approximation) when parameterized by $\mathbf{mod}_{\mathcal{H}}$. Hence, as corollaries of our meta-results, we also get the first FPT-ASes for the following (non-exhaustive list of) problems parameterized by $\mathbf{tw}_{\mathcal{H}}$ (and hence, also by $\mathbf{ed}_{\mathcal{H}}$). When $\mathcal{H}$ is apex-minor free, we obtain this result for INDEPENDENT SET, TRIANGLE PACKING (note that the latter is a special case of $\mathcal{F}$-SUBGRAPH PACKING). When $\mathcal{H}$ is minor free, we are able to infer the same result for VERTEX COVER, FEEDBACK VERTEX SET, CYCLE PACKING (and more generally, $\mathcal{F}$-MINOR PACKING). 5

Finally, we identify DOMINATING SET, CONNECTED DOMINATING SET, and CONNECTED VERTEX COVER as interesting special cases that are not covered by the framework developed above and give purpose-built FPT-ASes for them parameterized by $\mathbf{mod}_{\mathcal{H}}$ and then also prove an equivalence theorem in the same spirit as Theorem 1. Among these three problems, we give the details for DOMINATING SET, and defer the results for the connectivity constrained problems to the full version of the paper.

In summary, our work highlights many natural problems for which the answer to Question 2 is affirmative. Finally, we note that in Theorem 1 and its variants, due to their generality, we only obtain *non-uniform* FPT-ASes. In Conclusion, we briefly discuss certain scenarios where it may be possible to obtain uniform FPT-ASes.

1.2 Our techniques

We first summarize the technique behind our FPT-ASes parameterized by $\mathbf{mod}_{\mathcal{H}}$, which we dub "Bucket vs Ocean". Recall that this is required in order to invoke Theorem 1 to get our eventual results, i.e., FPT-ASes parameterized by $\mathbf{tw}_{\mathcal{H}}$ and $\mathbf{ed}_{\mathcal{H}}$. We remark that throughout the paper, we assume that our FPT-AS parameterized by, say $\mathbf{mod}_{\mathcal{H}}$ (resp. $\mathbf{ed}_{\mathcal{H}}, \mathbf{tw}_{\mathcal{H}}$) is also provided with a modulator to $\mathcal{H}$ (resp. $\mathcal{H}$-elimination/tree decomposition) of the appropriate size (resp. width). For $\mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$, one can use algorithms from [9, 15, 1, 14] to compute the appropriate decompositions exactly. Alternatively, one can use the recent result of [16] to compute constant approximations thereof.

Consider a special case of VERTEX DELETION TO $\mathcal{F}$, say VERTEX COVER. Consider an H-minor free graph family $\mathcal{H}$, where VC admits an EPTAS. Consider a graph G and $M \subseteq V(G)$ of size $\mathbf{mod}_{\mathcal{H}}(G)$ such that $G - M \in \mathcal{H}$. We use the known EPTAS on $G - M$ to compute an $(1 + \epsilon/2)$-approximate solution to VC, call it S. Next, we compare $|S|$ with $|M|$, and consider two cases. If $|M| < \epsilon/3 \cdot |S| \leq (1 + \epsilon/2) \cdot \mathsf{OPT}(G - M)$, i.e., $\mathsf{OPT}(G - M)$ is like an "ocean", compared to a "bucket" of water that is M, then we can add the bucket to the ocean "for free", i.e., $|S| + |M| \leq (1+\epsilon) \cdot \mathsf{OPT}(G - M) \leq (1+\epsilon) \cdot \mathsf{OPT}(G)$. Otherwise, if $|M|$ and $|S|$ are comparable, then $\mathsf{OPT}(G) \leq 3\mathbf{mod}_{\mathcal{H}}(G)/\epsilon$, in which case we can use the FPT algorithm for Vertex Cover, which is in fact in FPT in $\mathbf{mod}_{\mathcal{H}}$ and ϵ. This is the high level idea behind our FPT-ASes parameterized by $\mathbf{mod}_{\mathcal{H}}$, although one has to overcome several problem-specific challenges to make it work for the other problems.

A more sophisticated version of the idea also turns out to be useful in proving the equivalence theorem. Again, let us consider the example of Vertex Cover and $\mathcal{H}$ being an H-minor free graph family. Having armed ourselves with an FPT-AS parameterized by $\mathbf{mod}_{\mathcal{H}}$ as in the previous paragraph, now our task is to generalize to $\mathbf{tw}_{\mathcal{H}}$. To this end, we consider each bag $\chi(t)$ corresponding to a node t in the $\mathcal{H}$-tree decomposition (defined formally in the next section), which consists of $\ell \leq \mathbf{tw}_{\mathcal{H}}(G)+1$ vertices, say R_t, which locally act like a modulator to a disjoint "base graph", say $G[H_t] \in \mathcal{H}$. We use the same "Bucket vs Ocean" idea to classify each node as *good* or *bad*. A node t is *good*, if a $(1+\epsilon)$-approximate solution S_t for $G[H_t]$ is like an "ocean" compared to R_t. In this case, we can again add R_t to S_t "for free". Otherwise, we say that a node t is bad, if $\mathsf{OPT}(G[H_t])$ is bounded by $3\ell/\epsilon$. From here, our task is to reduce the treewidth of the graph induced by the vertices in the bags of bad nodes. To this end, we use the FII property as well as the FPT-AS to perform a series of "graph replacements" in each bad node, where in each iteration, we replace the corresponding bag with an "equivalent subgraph" of size bounded by a function of ℓ and ϵ. At the end of this procedure, the resulting graph has treewidth bounded by a function of ℓ and ϵ, and we can use the CMSO-definability and Courcelle's theorem [3] to find an optimal solution (that can then be translated back to the original instance). It can be shown that the resulting solution is a $(1+\epsilon)$-approximate solution for the entire graph, and the whole algorithm runs in time FPT in $\mathbf{tw}_{\mathcal{H}}$ and ϵ.

Due to space constraints some of the results and figures are moved to appendix and can be found in the full version of the paper [13].

2 Preliminaries

For an instance $\mathcal{I}$ of an optimization problem Π, we denote the value/size of an optimal solution by $\mathsf{OPT}_{\Pi}(\mathcal{I})$, and may omit the subscript if the problem is clear from the context. An algorithm that, for any feasible instance of minimization (resp. maximization) problem Π, returns a solution of cost/size at most (resp. at least) $\alpha \cdot \mathsf{OPT}$, is called an α-approximation. We say that a minimization (resp. maximization) problem Π admits an FPT-AS (FPT-Approximation Scheme) parameterized by a parameter t and ϵ, if there exists an $(1+\epsilon)$-approximation (resp. $(1-\epsilon)$-approximation) algorithm with running time of the form $f(t,\epsilon) \cdot |\mathcal{I}|^{\mathcal{O}(1)}$, for some computable function f. Similarly, we say that Π admits an EPTAS (efficient polynomial-time approximation scheme), if there exists a $(1+\epsilon)$-approximation (resp. a $(1-\epsilon)$-approximation) algorithm with running time of the form $g(\epsilon) \cdot |\mathcal{I}|^{\mathcal{O}(1)}$, for some computable function g.

Let $\mathcal{G}$ denote the family of all graphs. We say that a family $\mathcal{F} \subseteq \mathcal{G}$ of graphs is *hereditary* if for any graph $G \in \mathcal{F}$, every induced subgraph G' of G also belongs to $\mathcal{F}$. We say that a family $\mathcal{F} \subseteq \mathcal{G}$ is *closed under disjoint union* if for any $G_1, G_2 \in \mathcal{F}$, the graph G obtained by taking the disjoint union of G_1 and G_2, also belongs to $\mathcal{F}$.

Graph Decompositions

The following definitions are borrowed from [1]. For a graph G, $\mathbf{mod}_{\mathcal{H}}(G)$ denotes the size of a smallest vertex set S such that $G - S \in \mathcal{H}$. If $G - S \in \mathcal{H}$, then S is called a *modulator* to $\mathcal{H}$.

▶ **Definition 2.** *For a graph family $\mathcal{H}$, an $\mathcal{H}$-*elimination decomposition *of G is a triple (T, χ, L), where T is a rooted forest, $\chi : V(T) \to 2^{V(G)}$, and $L \subseteq V(G)$ such that:*

1. *For each internal node $t \in V(T)$, we have that $|\chi(t)| \leq 1$, and $\chi(t) \subseteq V(G) \setminus L$.*
2. *The sets $(\chi(t))_{t \in V(T)}$ form a partition of $V(G)$.*
3. *For each leaf t in T, we have $\chi(t) \subseteq L$, such that the graph $G[\chi(t)]$, called a base component, belongs to $\mathcal{H}$. Furthermore, $(\chi(t))_{leaf\ t}$ forms a partition of L.*
4. *For each edge $uv \in V(G)$, if $u \in \chi(t_1)$, and $v \in \chi(t_2)$, then t_1 and t_2 are in ancestor-descendant relation in T.*

The depth of a rooted tree T is the maximum number of edges on a root-to-leaf path in T. We refer to the union of base components as the set of base vertices. The $\mathcal{H}$-elimination distance of G, denoted as $\mathbf{ed}_{\mathcal{H}}(G)$, is the minimum depth of an $\mathcal{H}$-elimination forest for G. Note that a pair (T, χ) is a (standard) elimination forest of $\mathcal{H}$ is a class of empty graphs, i.e., the base components are empty. In this case, $\mathbf{ed}_{\mathcal{H}}(G)$ is known as the *treedepth* of G, and is denoted as $\mathbf{td}(G)$.

Just like the notion of $\mathcal{H}$-elimination decomposition generalizes the notion of elimination forest, the following is an analogous generalization of the notion of tree decomposition.

▶ **Definition 3.** *For a graph family $\mathcal{H}$, an $\mathcal{H}$-*tree decomposition *of a graph G is a triple (T, χ, L), where T is a rooted tree, $\chi : V(T) \to 2^{V(G)}$, and $L \subseteq V(G)$, such that:*

1. *For each $v \in V(G)$, the nodes $\{t \in V(T) : v \in \chi(t)\}$ induce a non-empty connected subtree in T.*
2. *For each edge $uv \in E(G)$, there is a node $t \in V(G)$ with $\{u, v\} \in \chi(t)$.*
3. *For each vertex $v \in L$, there is a* unique leaf *$t \in V(T)$ for which $v \in \chi(t)$.*
4. *For each leaf node $t \in V(T)$, the graph $G[\chi(t) \cap L] \in \mathcal{H}$.*

The *width* of an $\mathcal{H}$-tree decomposition is defined as $\max\left\{0, \max_{t \in V(T)} |\chi(t) \setminus L| - 1\right\}$. The $\mathcal{H}$-treewidth of G, denoted by $\mathbf{tw}_{\mathcal{H}}(G)$, is the minimum width of an $\mathcal{H}$-tree decomposition of G. The connected components of $G[L]$ are called base components, and the vertices in L are called base vertices.

A pair (T, χ) is a (standard) *tree decomposition* if $(T, \chi, \emptyset)$ satisfies all conditions of an $\mathcal{H}$-decomposition, where the choice of $\mathcal{H}$ is irrelevant.

Additional preliminaries can be found in the full version of the paper.

3 Vertex Deletion to $\mathcal{F}$

In this section we give our "first equivalence" result. Towards that, first in Section 3.2, we design concrete FPT-ASes for several Vertex Deletion to $\mathcal{F}$ type problems (e.g., Vertex Cover, Feedback Vertex Set parameterized by $\mathbf{mod}_{\mathcal{H}}$, where $\mathcal{H}$ is an apex-minor family of graphs. Then, in Section 3.3 we prove our main equivalence theorem, which lets us extend the previous FPT-ASes parameterized by $\mathbf{mod}_{\mathcal{H}}(\cdot)$ to $\mathbf{ed}_{\mathcal{H}}(\cdot)$, and $\mathbf{tw}_{\mathcal{H}}(\cdot)$.

3.1 Preliminaries for Vertex Deletion to $\mathcal{F}$

We focus on the following problem, specifically on the case where $\mathcal{F}$ is some fixed well-behaved family of graphs.

In Vertex Deletion to $\mathcal{F}$ problem, we are given an instance (G, k), where $G = (V, E)$ is a graph, and $k \geq 0$ is an integer. And the question is does there exist a subset $S \subseteq V(G)$ of size at most k, such that $G - S \in \mathcal{F}$?

Optimization variant

Let $\mathcal{F}$ be a well-behaved family of graphs. For a graph G, a set $S \subseteq V(G)$ is said to be a *solution* to Vertex Deletion to $\mathcal{F}$, if $G - S \in \mathcal{F}$. In the optimization variant of Vertex Deletion to $\mathcal{F}$, we want to find a solution of the smallest cardinality. By slightly abusing the notation, we will use Vertex Deletion to $\mathcal{F}$ to refer to the decision as well as the optimization version, and will only disambiguate when strictly necessary. Note that assuming $\mathcal{F}$ contains at least one graph, the hereditary property implies that the empty graph belongs to $\mathcal{F}$, which means that $V(G)$ is always a solution to Vertex Deletion to $\mathcal{F}$. The proof of the following observation is straightforward and can be found in the full version.

▶ **Observation 4.** *Let $\mathcal{F}$ be a well-behaved family, and let $\Pi =$ Vertex Deletion to $\mathcal{F}$. Then, for any graph G and any subset $S \subseteq V(G)$, it holds that*

$$\mathsf{OPT}_\Pi(G - S) \leq \mathsf{OPT}_\Pi(G) \leq \mathsf{OPT}_\Pi(G - S) + |S|.$$

We also note that Vertex Deletion to $\mathcal{F}$ is *self-reducible*, i.e., an algorithm for the decision version can be used to actually find a solution set. We define this notion formally and give the proof in the full version.

3.2 FPT-ASes for Vertex Deletion to $\mathcal{F}$ Parameterized by $\mathbf{mod}_\mathcal{H}(\cdot)$

Let $\mathcal{H}, \mathcal{F}$ be well-behaved families of graphs, such that $\Pi =$ Vertex Deletion to $\mathcal{F}$ admits an EPTAS for (the optimization variant of) Π on any $G \in \mathcal{H}$. Then, we show how to design an FPT-AS for Π parameterized by $\mathbf{mod}_\mathcal{H}(\cdot)$, and ϵ. Note that, for a graph G, when $p := \mathbf{mod}_\mathcal{H}(G) = 0$, and in this case our algorithm is essentially an EPTAS. In this sense, we generalize the assumed EPTAS on $\mathcal{H}$ to a larger family of graphs that are at most p vertices away from $\mathcal{H}$.

Let $M \subseteq V(G)$ be a modulator to $\mathcal{H}$ of size p, i.e., $G - M =: F \in \mathcal{H}$. We first compute a $(1 + \epsilon/2)$-approximate solution S for F using EPTAS in time $g(\epsilon) \cdot n^{\mathcal{O}(1)}$. Note that $S \subseteq V(G) \setminus M$, and $p \leq (1 + \epsilon/2) \cdot \mathsf{OPT}(F) \leq (1 + \epsilon/2)\mathsf{OPT}(G)$, where the last inequality follows from Observation 4. Depending on the relative sizes of M, and the approximate solution S, we consider the following two cases.

1. $p \leq \frac{\epsilon}{3} \cdot |S|$.
   Then we simply return $X \cup S$ as a solution for G. Note that $|M| + |S| \leq (1 + \epsilon/3)(1 + \epsilon/2) \cdot \mathsf{OPT}(F) \leq (1 + \epsilon) \cdot \mathsf{OPT}(F) \leq (1 + \epsilon) \cdot \mathsf{OPT}(G)$.
2. $p > \frac{\epsilon}{3} \cdot |S|$, i.e, $\mathsf{OPT}(F) \leq |S| \leq \frac{3p}{\epsilon}$.
   Then, by Observation 4, $\mathsf{OPT}(G) \leq \mathsf{OPT}(F) + |M| \leq p + \frac{3p}{\epsilon} =: p'$. In this case, we use an FPT algorithm parameterized by the solution size, to find an optimal solution for G. Note that this takes $h(p') \cdot n^{\mathcal{O}(1)}$, i.e., $f(p, \epsilon) \cdot n^{\mathcal{O}(1)}$ time, where $f(p, \epsilon) = h(p + 3p/\epsilon)$.

Thus, we get the following theorem.

▶ **Theorem 5.** *Let $\mathcal{F}, \mathcal{H}$ be well-behaved families of graphs. Moreover, suppose that* $\Pi =$ Vertex Deletion to $\mathcal{F}$ *admits an* EPTAS *on any graph in $\mathcal{H}$, and that Π is* FPT *parameterized by the solution size. Then, there exists an* FPT-AS *for Π, parameterized by $p \coloneqq \mathbf{mod}_{\mathcal{H}}(G)$ with running time $\max\{g(\epsilon), f(p, \epsilon)\} \cdot n^{\mathcal{O}(1)}$.*

Corollaries

Let Π be either Vertex Cover or Feedback Vertex Set. Note that Π is Vertex Deletion to $\mathcal{F}$, where $\mathcal{F}$ is family of isolated vertices, and forests, respectively. If $\mathcal{H}$ is a family of apex-minor free graphs, then Π admits an EPTAS on $\mathcal{H}$ with running time $2^{\mathcal{O}(1/\epsilon)} \cdot n^{\mathcal{O}(1)}$ ([7], Corollary 2). Furthermore, Π is FPT parameterized by solution size on general graphs, with running time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$, where k denotes the solution size. Therefore, we get an FPT-AS parameterized by $p \coloneqq \mathbf{mod}_{\mathcal{H}}(G)$ with running time $2^{\mathcal{O}(p/\epsilon)} \cdot n^{\mathcal{O}(1)}$ for Vertex Cover and Feedback Vertex Set.

We note that it is possible to improve the running time of FPT-AS for Vertex Cover using an even simpler algorithm – we simply guess the intersection of an optimal solution with X by iterating over all subsets of X. Let $Y \subseteq X$ be a guess. If Y is feasible, then it must be that $U \coloneqq X \setminus Y$ must be an independent set, and $M \coloneqq N(U) \cap (V(G) \setminus X)$ must belong to the solution. Thus, the remaining graph is $G \setminus (M \cup X)$, which belongs to the family $\mathcal{H}$. Then, we use the $2^{\mathcal{O}(1/\epsilon)} \cdot n^{\mathcal{O}(1)}$ time EPTAS on $\mathcal{H}$ to obtain a solution S and return the smallest possible solution of the form $Y \cup M \cup S$. Thus, we get a $2^{\mathcal{O}(\mathbf{mod}_{\mathcal{H}}(G)+1/\epsilon)} \cdot n^{\mathcal{O}(1)}$ time FPT-AS.

▶ **Corollary 6.** *There exists an FPT-AS for* Vertex Cover *(resp.* Feedback Vertex Set*) that runs in time $2^{\mathcal{O}(p+\frac{1}{\epsilon})} \cdot n^{\mathcal{O}(1)}$ (resp. $2^{\mathcal{O}(p/\epsilon)} \cdot n^{\mathcal{O}(1)}$), where $p = \mathbf{mod}_{\mathcal{H}}(G)$, and $\mathcal{H}$ is a family of apex-minor free graphs.*

More generally, this result extends for Planar $\mathcal{R}$-Deletion, which is a special case of Vertex Deletion to $\mathcal{F}$, where the family $\mathcal{F}$ is characterized by a set $\mathcal{R}$ of forbidden minors containing at least one planar graph. We note that Fomin et al. [7] give an EPTAS for this problem on H-minor free graphs. On the other hand, this problem is known to admit a single-exponential FPT algorithm parameterized by the solution size via the results from Fomin et al. [6]. Thus, we get the following corollary.

▶ **Corollary 7.** *There exists an* FPT-AS *for* Planar $\mathcal{R}$-Deletion*, that runs in time $f(p, \epsilon) \cdot n^{\mathcal{O}(1)}$, where $p = \mathbf{mod}_{\mathcal{H}}(G)$, and $\mathcal{H}$ is a family of H-minor free graphs, for some fixed graph H and some computable function f. In particular, this implies an* FPT-AS *for* Treewidth-η-Modulator.[1]

Finally, we mention Odd Cycle Transversal. Note that there exists a polynomial time $\frac{9}{4}$-approximation for Odd Cycle Transversal on planar graphs [11], and there exists a $3^k n^{\mathcal{O}(1)}$ time algorithm for the problem parameterized by the solution size [20]. Thus, by using the 9/4-approximation instead of an $(1+\epsilon/2)$-approximation on $G \setminus X$, followed by a similar case analysis, we get the following result.

▶ **Corollary 8.** *There exists an algorithm that runs in time $2^{\mathcal{O}(p/\epsilon)} \cdot n^{\mathcal{O}(1)}$, where p is the size of planar vertex deletion set, to compute a $(\frac{9}{4}+\epsilon)$-approximation for* Odd Cycle Transversal.

[1] For a fixed $\eta \geq 0$, Treewidth-η-Modulator is the problem of deciding whether one can delete at most k vertices from a given graph G, such that the resulting graph has treewidth at most η.

3.3 Equivalence Theorem

In this section we obtain the following equivalence statement.

▶ **Theorem 1.** *Let $\mathcal{H}, \mathcal{F}$ be well-behaved families of graphs, where $\mathcal{F}$ is CMSO-definable. Suppose* $\Pi =$ Vertex Deletion to $\mathcal{F}$ *has* FII. *Then, the following statements are equivalent.*

1. Π *admits an* FPT-AS *parameterized by* $\mathbf{mod}_{\mathcal{H}}(\cdot)$ *and* ϵ.
2. Π *admits an* FPT-AS *parameterized by* $\mathbf{ed}_{\mathcal{H}}(\cdot)$ *and* ϵ.
3. Π *admits an* FPT-AS *parameterized by* $\mathbf{tw}_{\mathcal{H}}(\cdot)$ *and* ϵ.

Proof. To prove that the three items are equivalent, we first note that for any graph G and any well-behaved family $\mathcal{H}$, it holds that $\mathbf{mod}_{\mathcal{H}}(G) \geq \mathbf{ed}_{\mathcal{H}}(G) \geq \mathbf{tw}_{\mathcal{H}}(G)$. Therefore, $3 \Longrightarrow 2 \Longrightarrow 1$. Thus, we focus on proving $1 \Longrightarrow 3$. Let $\mathcal{P}_{\mathbf{mod}}$ denote the assumed FPT-AS for Π parameterized by $\mathbf{mod}_{\mathcal{H}}$ and ϵ. That is, for any graph G on n vertices, $\mathcal{P}_{\mathbf{mod}}$ runs in time $f(\mathbf{mod}_{\mathcal{H}}(G), \epsilon) \cdot n^{\mathcal{O}(1)}$, and returns a subset $S \subseteq V(G)$ such that (i) $G - S \in \mathcal{F}$, and (ii) $|S| \leq (1+\epsilon) \cdot \mathsf{OPT}_{\Pi}(G)$. [2].

Let $G = (V, E)$ be a graph, and let (T, χ, L) be the given $\mathcal{H}$-tree decomposition of G of width at most $\mathbf{tw}_{\mathcal{H}}(G)$, and let $\ell \coloneqq \mathbf{tw}_{\mathcal{H}}(G)$ (resp. $\ell \coloneqq \mathbf{ed}_{\mathcal{H}}(G)$). Recall that $\mathcal{P}_{\mathbf{mod}}$ returns a $(1+\epsilon)$-approximation to Π on any instance (G, k) of Π.

For every leaf node $t \in V(T)$, let $V_t \coloneqq \chi(t)$, and let $H_t = V_t \cap L$ and $R_t = V_t \setminus L$. The properties of $\mathcal{H}$-tree decomposition imply that $G[H_t] \in \mathcal{H}$, and $|R_t| \leq \ell$. Equivalently, $R_t \subseteq V(G) \setminus L$ is a modulator to $\mathcal{H}$ for the graph $G[\chi(t)]$. Note that the base vertices, i.e., the vertices in L, are partitioned across the leaf bags, which implies that the graphs $\{G[H_t]\}_{\text{leaf } t \in V(T)}$ are disjoint.

We iterate over every leaf nodes $t \in V(T)$, and proceed as follows. Note that $G[H_t] \in \mathcal{H}$, thus $G[H_t]$ has a modulator of size zero to the family $\mathcal{H}$. Therefore, we can use $\mathcal{P}_{\mathbf{mod}}$ to obtain a set $S_t \subseteq H_t$ that is a $(1+\epsilon/2)$-approximation for Π on the graph $G[H_t]$, i.e., $|S_t| \leq (1+\epsilon/2) \cdot \mathsf{OPT}(G[H_t])$. This takes $f(0, \epsilon/2) \cdot |H_t|^{\mathcal{O}(1)} \leq h(\epsilon) \cdot n^{\mathcal{O}(1)}$ time for some function $h(\cdot)$. In other words, $\mathcal{P}_{\mathbf{mod}}$ is an EPTAS for Π on $G[H_t]$.

For every leaf node $t \in V(T)$, if $|R_t| \leq \epsilon/3 \cdot |S_t|$, then we say that the node is *good*. Otherwise, we say that $t \in V(T)$ is *bad*. Furthermore, all non-leaf nodes are classified as *bad*. Note that for a good node $t \in V(T)$, $|R_t| \leq (1+\epsilon/2) \cdot (\epsilon/3) \cdot \mathsf{OPT}(G[H_t]) \leq (\epsilon/2) \cdot \mathsf{OPT}(G[H_t])$. On the other hand, for a bad node $t \in V(T)$, $\ell \geq |R_t| \geq \epsilon/3 \cdot |S_t| \geq \epsilon/3 \cdot \mathsf{OPT}(G[H_t])$. Let U_g and U_b denote the set of good and bad nodes respectively. Let $S_1 = \bigcup_{t \in U_g} S_t$, and $S_2 = \bigcup_{t \in U_g} R_t$.

Furthermore, let $V_g = \bigcup_{t \in U_g} \chi(t)$, and $V_b = V \setminus V_g$. Now, let $S_b \subseteq V_b$ denote an $(1+\epsilon)$-approximate solution to Π on the graph $G[V_b]$. We first state the following lemma, whose proof is given after the current proof.

▶ **Lemma 9.** $|S| \leq (1+\epsilon) \cdot \mathsf{OPT}(G)$, *where* $S = S_1 \cup S_2 \cup S_b$ *as defined above.*

Let $F = G[V_b]$. From the $\mathcal{H}$-tree decomposition (T, χ, L) of G of width ℓ, it is possible to obtain a $\mathcal{H}$-tree decomposition of F of width at most ℓ, by deleting the vertices of V_g from every bag. For simplicity, we will continue to use (T, χ, L) for the new "projected" $\mathcal{H}$-tree decomposition of F. As suggested by Lemma 9, we need to find a set S_b, which is an $(1+\epsilon)$-approximation to Π on $F \coloneqq G[V_b]$. In fact, we will show how to find an optimal solution. To this end, we rely on the following technical lemma.

[2] Note that $\mathcal{P}_{\mathbf{mod}}$ may in fact be a family of approximation algorithms, containing an algorithm for each value of $\mathbf{mod}_{\mathcal{H}}(\cdot)$ and ϵ. We slightly abuse the notation and assume that we use an appropriate algorithm from this family, based on the value of $\mathbf{mod}_{\mathcal{H}}$ of the relevant graph, and ϵ.

▶ **Lemma 10.** *Let $\mathcal{H}, \mathcal{F}$ be well-behaved families of graphs, where $\mathcal{F}$ is CMSO-definable. Suppose that $\Pi =$ Vertex Deletion to $\mathcal{F}$ has* FII*. Furthermore, suppose there exists an* FPT-AS $\mathcal{P}_{\mathbf{mod}}$ *for Π, parameterized by $\mathbf{mod}_{\mathcal{H}}(\cdot)$ and ϵ. Then, for every positive integer ℓ, there exists an algorithm $\mathcal{A}_\ell$, that takes input (G, T, χ, L), where*

1. *G is a graph on at most n vertices, and*
2. *A $\mathcal{H}$-tree decomposition (T, χ, L) of G of width ℓ, such that:*
   $\star$ *for every leaf $t \in V(T)$, the graph $H_t := G[\chi(t) \cap L]$ satisfies $\mathsf{OPT}_\Pi(H_t) \leq 3\ell/\epsilon$.*

The algorithm $\mathcal{A}_\ell$ runs in time $g(\ell, \epsilon) \cdot n^{\mathcal{O}(1)}$, where g is some function, and returns an optimal solution $S \subseteq V(G)$ to Π on G, i.e., $|S| = \mathsf{OPT}_\Pi(G)$, and $G - S \in \mathcal{F}$.

The idea of the proof of Lemma 10 is similar to that of Theorem 6.1 from [1]. However, they require that $\mathcal{P}_{\mathbf{mod}}$ be an *exact algorithm* for Π. Our result builds upon the fact that, when the input graph satisfies the property $\star$, an *approximation algorithm* can be made to behave like an exact algorithm by a suitable choice of ϵ. However, a formal proof requires several technical definitions pertaining to FII, and thus is deferred to the full version of the paper.

Now we finish the proof of Theorem 1. Note that the graph $G[V_b]$ and the $\mathcal{H}$-tree decomposition (T, χ, L) restricted to $G[V_b]$ satisfies the requirements of Lemma 10 by construction (recall that $V_b =$. Thus, using the algorithm from Lemma 10, we can, in fact, get an optimal solution $S_b \subseteq V_b$ to Π on $G[V_b]$. This concludes the proof of Theorem 1, modulo the proof of Lemma 9. ◀

Proof of Lemma 9. Recall the definitions of the partial solutions S_1, S_2, S_b as defined in the proof of Theorem 1. From Observation 4, we have the following inequality.

$$\mathsf{OPT}(G - S_2) \leq \mathsf{OPT}(G) \tag{1}$$

Note that the properties of $\mathcal{H}$-tree decomposition imply that, if we delete S_2 from G, the graph $G - S_2$ consists of disjoint induced subgraphs H_t for $t \in U_g$, as well as $G[V_b]$ (note that each of these induced subgraphs may or may not be connected). Now, consider

$$\begin{aligned}
|S| &= |S_1| + |S_2| + |S_b| && \text{(Since } S = S_1 \uplus S_2 \uplus S_b.\text{)}\\
&\leq \sum_{t \in U_g} (|S_t| + |R_t|) + |S_b| && \text{(Since } S_1 = \textstyle\bigcup_{t \in U_g} S_t \text{ and } S_2 = \bigcup_{t \in U_g} R_t\text{)}\\
&\leq \left(\sum_{t \in U_g} (1 + \epsilon/2) \cdot \mathsf{OPT}(G[H_t]) + \epsilon/2 \cdot \mathsf{OPT}(G[H_t]) \right) + (1 + \epsilon) \cdot \mathsf{OPT}(G[V_b])\\
& && \text{(Using bounds on } S_t, R_t \text{ and } S_b \text{ respectively)}\\
&\leq (1 + \epsilon) \cdot \left(\left(\sum_{t \in U_g} \mathsf{OPT}(G[H_t]) \right) + \mathsf{OPT}(G[V_b]) \right)\\
&\leq (1 + \epsilon) \cdot \mathsf{OPT}(G) && \text{(since } V(G) \setminus S_2 = V_b \uplus \biguplus_{t \in U_g} H_t \text{, and (1))}
\end{aligned}$$

◀

3.4 Variations of the Equivalence Theorem

Theorem 1 gives non-uniform FPT-ASes. In the following, we obtain a simpler, and *explicit* version of Theorem 1 for Vertex Deletion to $\mathcal{F}$ when $\mathcal{F}$ satisfies the following property, called *treewidth-η-modulated*: Suppose there exists a constant η, such that the (standard) treewidth of every graph in $\mathcal{F}$ is bounded by η. In this case, we say that the family $\mathcal{F}$ is *treewidth-η-modulated*, or simply, *η-modulated*. Note that many natural VERTEX DELETION TO $\mathcal{F}$ problems are η-modulated. For example, VERTEX COVER (resp. FEEDBACK VERTEX SET) is η-modulated for $\eta = 0$ (resp. $\eta = 1$). More generally, it is also known if the family $\mathcal{F}$ is characterized by a family of excluded minors $\mathcal{O}$ containing at least one planar graph, then VERTEX DELETION TO $\mathcal{F}$ is known as PLANAR $\mathcal{O}$-DELETION, and is also η-modulated for some η that depends on $\mathcal{O}$. The proof of this theorem is straightforward given the η-modulated property of $\mathcal{F}$. A formal proof can be found in the full version.

▶ **Theorem 11.** *Let $\mathcal{H}, \mathcal{F}$ be well-behaved families of graphs. Moreover, suppose that Π = VERTEX DELETION TO $\mathcal{F}$ is such that: (i) $\mathcal{F}$ is η-modulated for some constant $\eta \geq 0$, and (ii) there exists an exact FPT algorithm for Π parameterized by $\mathbf{tw}(\cdot)$. Then, the following statements are equivalent.*
1. *Π admits an FPT-AS parameterized by $\mathbf{mod}_{\mathcal{H}}(\cdot)$ and ϵ.*
2. *Π admits an FPT-AS parameterized by $\mathbf{ed}_{\mathcal{H}}(\cdot)$ and ϵ.*
3. *Π admits an FPT-AS parameterized by $\mathbf{tw}_{\mathcal{H}}(\cdot)$ and ϵ.*

Note that Theorem 11 requires that Π admit an FPT-AS parameterized by $\mathbf{mod}_{\mathcal{H}}(G)$. In particular, when $\mathbf{mod}_{\mathcal{H}}(G) = 0 \Longleftrightarrow G \in \mathcal{H}$, Π should have an EPTAS. However, in some cases, we only know an α-approximation for Π even when $G \in \mathcal{H}$, where $\alpha \geq 1$. In such cases, the following theorem may be applicable, if, Π is also known to be FPT parameterized by the size of the solution.

▶ **Theorem 12.** *Let $\mathcal{H}, \mathcal{F}$ be well-behaved families of graphs. Moreover, suppose that Π = VERTEX DELETION TO $\mathcal{F}$ satisfies the following properties: (i) Π is CMSO-definable, (ii) has FII, and (iii) Π is FPT parameterized by the size of the solution. Also suppose that for some constant $\alpha \geq 1$, Π admits an α-approximation in time $f(\mathbf{mod}_{\mathcal{H}}(\cdot)) \cdot n^{\mathcal{O}(1)}$. Then,*
1. *Π admits an $(\alpha + \epsilon)$-approximation in time $g(\mathbf{ed}_{\mathcal{H}}(\cdot), \epsilon) \cdot n^{\mathcal{O}(1)}$, and*
2. *Π admits an $(\alpha + \epsilon)$-approximation in time $h(\mathbf{tw}_{\mathcal{H}}(\cdot), \epsilon) \cdot n^{\mathcal{O}(1)}$.*

Plugging in Corollary 8 into Theorem 12, we obtain the following corollary.

▶ **Corollary 13.** *There exists an algorithm that runs in time $f(\ell, \epsilon) \cdot n^{\mathcal{O}(1)}$, where ℓ is $\mathbf{ed}_{\mathcal{H}}(G)$ (or $\mathbf{tw}_{\mathcal{H}}(G)$) to planar graphs, and computes a $(\frac{9}{4} + \epsilon)$-approximation for ODD CYCLE TRANSVERSAL.*

4 FPT-ASes for Dominating Set

In this section, we consider DOMINATING SET, defined as follows.

BLUE-WHITE DOMINATING SET
Input: An instance (G, k), where G is a graph, and k is a non-negative integer
Question: Does G contain a dominating set of size at most k, i.e., does there exist a set $S \subseteq V(G)$ such that for each $u \in V(G)$, $N[v] \cap S \neq \emptyset$?

Note that the main difficulty is that DOMINATING SET is not monotone, i.e., it does not necessarily hold that for any $S \subseteq V(G)$, $\mathsf{OPT}(G - S) \leq \mathsf{OPT}(G)$. Thus, the approach for VERTEX DELETION TO $\mathcal{F}$ does not immediately generalize, and the arguments for DOMINATING SET are technically more involved. Furthermore, it is not immediately clear that DOMINATING SET is *self-reducible*. Therefore, we need to rely on different *annotated versions* of DOMINATING SET to design approximation algorithms for the problem.

First, the following theorem that is proved using arguments involving bidimensionality. The formal proof of the theorem can be found in the full version in the appendix.

▶ **Theorem 14.** *Suppose we are given a graph G, and a set $M \subseteq V(G)$ of size p, such that $G' := G \setminus M \in \mathcal{H}$, where $\mathcal{H}$ is a family of apex-minor free graphs. Then, there exists an $f(p, \epsilon) \cdot n^{\mathcal{O}(1)}$ time algorithm to find a $(1 + \epsilon)$-approximation for* Dominating Set *on G, where $f(p, \epsilon) = \max \left\{ 2^{\mathcal{O}(p+\sqrt{\frac{p}{\epsilon}})}, \quad 2^{\mathcal{O}(1/\epsilon)} \right\}$.*

Next, we prove the equivalence theorem (cf. Theorem 15), which implies FPT-AS for Dominating Set, parameterized by $\mathbf{ed}_{\mathcal{H}}(\cdot)$ (resp. $\mathbf{tw}_{\mathcal{H}}(\cdot)$) and ϵ. The proof of this theorem is conceptually similar to that of Theorem 1; however, it is technically much more involved due to the reasons mentioned above.

4.1 Equivalence Theorem for Dominating Set

In this section, we prove the following equivalence theorem for Dominating Set, assuming that $\mathcal{H}$ be *well-behaved*, i.e., hereditary and closed under disjoint union.

▶ **Theorem 15.** *Let $\mathcal{H}$ be any well-behaved family of graphs, and let $\Pi =$ Dominating Set. Then, the following statements are equivalent.*
1. *Π admits an FPT-AS parameterized by $\mathbf{mod}_{\mathcal{H}}(\cdot)$ and ϵ.*
2. *Π admits an FPT-AS parameterized by $\mathbf{ed}_{\mathcal{H}}(\cdot)$ and ϵ.*
3. *Π admits an FPT-AS parameterized by $\mathbf{tw}_{\mathcal{H}}(\cdot)$ and ϵ.*

Proof. We prove 1$\Longrightarrow$3. Let $\mathcal{P}_{\mathbf{mod}}$ be the assumed FPT-AS for Π parameterized by $\mathbf{mod}_{\mathcal{H}}$ and ϵ. Let $G = (V, E)$ be the input graph on n vertices, along with an $\mathcal{H}$-tree decomposition (T, χ, L). Let ℓ denote the width of the given $\mathcal{H}$-tree decomposition. First, if $\mathcal{H}$ only contains the empty graph $(\emptyset, \emptyset)$, our goal is to prove that Dominating Set admits an FPT-AS parameterized by (standard) $\mathbf{tw}(\cdot)$. Indeed, since Dominating Set is FPT parameterized by $\mathbf{tw}(\cdot)$, we assume that $\mathcal{H}$ contains at least one non-empty graph. In particular, due to hereditary property, $\mathcal{H}$ must contain a graph on a single vertex. Furthermore, we assume that $\ell \geq 1$ – otherwise G is a disjoint union of graphs belonging to $\mathcal{H}$. This implies that $G \in \mathcal{H}$, in which case all three statements are trivially equivalent.

As before, for any leaf node $t \in V(T)$, let $V_t = \chi(t)$, $H_t = V_t \cap L$, and $R_t = V_t \setminus H_t$. Note that $G[H_t] \in \mathcal{H}$, and the sets $\{H_t\}_{t \in V(T)}$ are pairwise disjoint, and $|R_t| \leq \ell$ for any $t \in V(T)$.

We iterate over every leaf node $t \in V(T)$, and create a new graph $\widetilde{G}_t$ as follows. Let v^* be a new vertex that is not in V_t. Let $V'_t = V_t \cup \{v^*\}$. Let $E'_t = E(G[V_t]) \cup \widetilde{E}$, where $\widetilde{E} = \{v^* u : u \in R_t\}$. That is, the graph $\widetilde{G}_t$ is obtained by adding a new vertex v^* to the graph $G[V_t]$ and making it adjacent to all vertices of R_t.

Note that $\widetilde{G}_t - R_t = G[H_t] \uplus G^*$, where $G^* = (v^*, \emptyset)$ is an isolated component containing v^*. Now, $G[H_t] \in \mathcal{H}$, and $G^* \in \mathcal{H}$, which implies that $\widetilde{G}_t - R_t = G[H_t] \uplus G^* \in \mathcal{H}$, since $\mathcal{H}$ is closed under disjoint union. Now, we use $\mathcal{P}_{\mathbf{mod}}$ on $\widetilde{G}_t$ to obtain $\widetilde{S}_t \subseteq V(\widetilde{G}_t)$, such that (i) $\widetilde{S}_t$ is a dominating set for $\widetilde{G}_t$, and (ii) $|\widetilde{S}_t| \leq (1 + \epsilon/4) \cdot \mathsf{OPT}(\widetilde{G}_t)$. Let $S_t = \widetilde{S}_t \setminus \{v^*\}$.

For a leaf node $t \in V(T)$ if $\ell \leq \frac{\epsilon}{29} \cdot |S_t|$, then we say that t is a *good* node; and *bad* otherwise. All internal nodes are classified as bad. Let U_g, U_b denote the sets of good and bad nodes respectively. Let $V_g = \bigcup_{t \in U_g} H_t$, $V_m = \bigcup_{t \in U_g} R_t$, and $V_b = V(G) \setminus (V_g \cup V_m)$.

Let D^* be an optimal dominating set for G, and for any $t \in U_g$, let $D^*_{1,t} = H_t \cap D^*$, and $D^*_{2,t} = R_t \cap D^*$, and $D^*_t = D^*_{1,t} \cup D^*_{2,t}$. Let $D^*_1 = \bigcup_{t \in U_G} D^*_{1,t}$, and $D^*_2 = \bigcup_{t \in U_g} D^*_{2,t}$. Note that $\left\{D^*_{1,t}\right\}_{t \in U_g}$ is a partition of D^*_1, whereas a vertex in D^*_2 may belong to multiple $D^*_{2,t}$'s. Finally, let $D^*_b = D^* \cap V_b$. Note that $\{D^*_1, D^*_2, D^*_b\}$ is a partition of D^*.

Now, let $S_1 \coloneqq \bigcup_{t \in U_g} S_t$, and $S_2 \coloneqq \bigcup_{t \in U_g} R_t$. We wish to add $S_1 \cup S_2$ to our solution. Since each R_t is a separator between H_t and $G - \chi(t)$, we can delete V_g from the graph. However, we cannot simply delete $V_m = S_2$ from the graph – since the vertices in S_2 may already dominate a subset of vertices in V_b. Therefore, we need to "remember" that we have decided to add S_2 into our solution. To this end, we define an *annotated* version of Dominating Set, which we call Blue-White Dominating Set. The Blue-White Dominating Set problem is defined as follows.

Blue-White Dominating Set
Input: An instance (G, B, W, k), where G is a graph, $V(G) = B \uplus W$, and k is a non-negative integer
Question: Does there exist a subset $S \subseteq V(G)$ of size at most k such that (i) S is a dominating set for G, and (ii) $B \subseteq S$?

Let $F \coloneqq G[S_2 \uplus V_b]$. We want to find the smallest integer k', such that (F, S_2, V_b, k') is a yes-instance of Blue-White Dominating Set. Let S_b denote an $(1+\epsilon/2)$-approximate solution on this instance, i.e., (i) $S_b \subseteq S_2 \cup V_b$ is a dominating set for F, (ii) $S_2 \subseteq S_b$, and (iii) $|S_b| \le (1+\epsilon/2) \cdot \mathsf{OPT}'(F)$, where $\mathsf{OPT}'(F)$ denotes the size of the optimal solution for the Blue-White Dominating Set instance. We prove the following technical lemma, the proof of which is rather involved, and thus given in the full version.

▶ **Lemma 16.** $|S| = |S_1| + |S_2| + |S_b| \le (1+\epsilon) \cdot \mathsf{OPT}(G)$

Thus, our task is reduced to finding an approximate solution to Blue-White Dominating Set on (F, S_2, V_b, k'); in fact, we will find an exact solution. To do this, first we prove the following lemma, which shows that Blue-White Dominating Set can be reduced to the (standard) dominating problem on an auxiliary graph. The proof is straightforward, and is given in the full version.

▶ **Lemma 17.** *Let $\mathcal{I} = (G, B, W, k)$ be an instance of* Blue-White Dominating Set*. Let G' be the graph obtained by attaching $N = n^2$ distinct pendant vertices to each vertex in B, where $n = |V(G)|$. Let $\mathcal{I}' = (G', k)$ be the resulting instance of* Dominating Set*. Then $\mathcal{I}$ is a yes-instance of* Blue-White Dominating Set *iff $\mathcal{I}'$ is a yes-instance of* Dominating Set*.*

As suggested by Lemma 17, we create a graph F', as follows. For each $u \in S_2$, let P_u denote the set of $|V(F)|^2$ distinct pendant vertices attached to u. By slightly abusing the notation, let (T, χ, L) be the original $\mathcal{H}$-tree decomposition, restricted to the vertices of F. We modify this to obtain an $\mathcal{H}$-tree decomposition for F'. Consider a vertex $u \in S_2$, and note that $u \in R_t$ for some $t \in V(T)$. We pick one such arbitrary node t, and add a child t_u, which becomes a leaf in T. For this node t_u, we define its bag $\chi'(t_u) = P_u \cup \{u\}$, where P_u becomes the set of *base vertices* in $\chi'(t_u)$, and u is a non-base vertex in $\chi'(t_u)$. Note that $G[P_u]$ is a set of isolated vertices, and thus belongs to $\mathcal{H}$. For all original nodes $t \in V(T)$, we set $\chi'(t) = \chi(t)$. Finally, $L' = \bigcup_{u \in S_2} P_u$, and let T' be the resulting tree. It is easy to see that (T', χ', L') is a valid $\mathcal{H}$-tree decomposition for F' of width at most ℓ.

Now, we can prove an analogous version of Lemma 10 for the instance (F, k') of Dominating Set. It is known that Dominating Set is CMSO-definable and has FII [8]. We highlight the key differences required to perform replacement using the FII property of Dominating Set. Note that for a node that was earlier classified as *bad*, $\mathsf{OPT}(F[H_t]) = \mathsf{OPT}(G[H_t]) \le 29\ell/\epsilon$, which implies that $\mathsf{OPT}(F[\chi'(t)]) \le 29\ell/\epsilon + \ell \le 30\ell/\epsilon$. On the other hand, if t is a newly added leaf node then $\mathsf{OPT}(F[\chi'(t)]) \leq |R_t| \leq \ell$. This implies that, when we want to find a replacement for $G[\chi'(t)]$ using an application of FII, we want to decide instances of the form $F[\chi'(t)] \oplus Y$, where $|V(Y)|$ is upper bounded by some function of ℓ. Therefore, $\mathsf{OPT}(F[\chi'(t)] \oplus Y) \leq \mathsf{OPT}(F[\chi'(t)]) + |V(Y)|$, which is upper bounded by some function of ℓ and ϵ. Therefore, a version of Lemma 10 can solve *the decision version* of Blue-White Dominating Set.

Now, we discuss how to find a solution, using the algorithm for the decision version as an oracle. We try the following for the values of $k = |S_2|, |S_2| + 1, \ldots$, until the first time the decision version returns that Blue-White Dominating Set$(F, S_2, V(F) \setminus S_2, k)$ is a yes-instance. Let k^* denote this value.

We maintain a partial solution S_b, which is initialized to S_2. Consider a leaf node $t \in V(T)$, such that S_b does not already dominate all vertices of $\chi'(t) \cap L'$. We pick an arbitrary *original* vertex $u \in \chi'(t) \cap L'$, and use the algorithm to decide the instance $(F, S_b \cup \{u\}, V(F) \setminus (S_b \cup \{u\}), k^*)$ of Blue-White Dominating Set. If the algorithm returns no, then we conclude that u does not belong to a solution of size k^*. Otherwise, we update S_b by adding u to it, and proceed to the next iteration.

Now we discuss the technicalities when we use the reduction from Lemma 17 on the newly created Blue-White Dominating Set instance. In this reduction, we add $N = n^2$ distinct pendant vertices to u, and let F_u be the resulting graph. We add all these vertices to L' and to $\chi'(t)$. We also remove u from L, i.e., move it from H_t to R_t. Note that the resulting base graph is a disjoint union of $F'[H_t] - u$ and the N pendant vertices, each of which belong to $\mathcal{H}$ due to the well-behaved property. The size of the modulator increases by 1. However, since $\mathsf{OPT}(F[\chi(t)])$ is bounded by $29\ell/\epsilon$, therefore, we move at most $29\ell/\epsilon$ vertices from H_t to L_t. Thus, the width of the resulting $\mathcal{H}$-tree decomposition remains bounded by $\ell + 29\ell/\epsilon$. Furthermore, even for an incorrect choice of u, $\mathsf{OPT}(F'[\chi'(t)])$ remains bounded by a function of ℓ and ϵ. Thus, a version of Lemma 10 can still be used to obtain a solution S_b of size k^* to the original instance (F, S_2, V_b, k^*). This concludes the proof of the theorem. ◀

5 Conclusion

In this paper, we have initiated a systematic exploration of the impact that recently introduced "hybrid" graph parameters have on the existence of good approximations for fundamental graph problems. In fact, we have shown that as far as the task of obtaining an FPT-AS is concerned, for many problems, designing an FPT-AS parameterized by the largest of these, i.e., $\mathbf{mod}_{\mathcal{H}}$, is sufficient to obtain an FPT-AS parameterized by both $\mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$. This result gives an approximation analogue of recent equivalence obtained between these parameters in the exact algorithmic setting.

To demonstrate concrete applicability of our techniques, we first designed FPT-ASes for many classical graph problems parameterized by $\mathbf{mod}_{\mathcal{H}}$, where $\mathcal{H}$ is an apex/H-minor free graph family. Then, using our equivalence theorems, we are able to lift these FPT-ASes to $\mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$. At this point, we would like to highlight that, in several concrete applications of our equivalence theorems, the *non-uniform* FPT-ASes can be made uniform. For example, we believe that one can obtain uniform FPT-ASes parameterized by $\mathbf{mod}_{\mathcal{H}}, \mathbf{ed}_{\mathcal{H}}$ and $\mathbf{tw}_{\mathcal{H}}$ for problems such as Vertex Cover, Feedback Vertex Set, Independent Set, Dominating Set, Cycle Packing, where $\mathcal{H}$ is an apex-minor free graph family.

We conclude with two final remarks. Firstly, the assumption of CMSO-definability can be relaxed. In fact, we only require an EPTAS for the problem on graphs of bounded treewidth. This would then enable one to apply our framework to problems that are not

CMSO-definable but are known to have good approximations on the graph family under consideration. Secondly, the initial step of our algorithm, i.e., FPT-ASes parameterized by $\mathbf{mod}_{\mathcal{H}}$, can be made to work with other graph families, such as (Unit) Disk Graphs, using known EPTASes for a number of graph problems [7, 17].

References

1 Akanksha Agrawal, Lawqueen Kanesh, Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Deleting, eliminating and decomposing to hereditary classes are all FPT-equivalent. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1976–2004, 2022. doi:10.1137/1.9781611977073.79.

2 Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica*, 75(2):363–382, 2016. doi:10.1007/s00453-015-0045-3.

3 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

4 Erik D. Demaine, Timothy D. Goodrich, Kyle Kloster, Brian Lavallee, Quanquan C. Liu, Blair D. Sullivan, Ali Vakilian, and Andrew van der Poel. Structural rounding: Approximation algorithms for graphs near an algorithmically tractable class. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 37:1–37:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ESA.2019.37.

5 Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020. doi:10.3390/a13060146.

6 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-deletion: Approximation, kernelization and optimal FPT algorithms. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 470–479. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.62.

7 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Excluded grid minors and efficient polynomial-time approximation schemes. *J. ACM*, 65(2):10:1–10:44, 2018. doi:10.1145/3154833.

8 Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: theory of parameterized preprocessing.* Cambridge University Press, 2019.

9 Robert Ganian, M. S. Ramanujan, and Stefan Szeider. Combining treewidth and backdoors for CSP. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPIcs*, pages 36:1–36:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.STACS.2017.36.

10 M. R. Garey and David S. Johnson. The rectilinear steiner tree problem is NP complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977.

11 Michel X. Goemans and David P. Williamson. Primal-dual approximation algorithms for feedback problems in planar graphs. *Comb.*, 18(1):37–59, 1998. doi:10.1007/PL00009810.

12 Jiong Guo, Falk Hüffner, and Rolf Niedermeier. A structural view on parameterizing problems: Distance from triviality. In Rodney G. Downey, Michael R. Fellows, and Frank K. H. A. Dehne, editors, *Parameterized and Exact Computation, First International Workshop, IWPEC 2004, Bergen, Norway, September 14-17, 2004, Proceedings*, volume 3162 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 2004. doi:10.1007/978-3-540-28639-4_15.

13 Tanmay Inamdar, Lawqueen Kanesh, Madhumita Kundu, M. S. Ramanujan, and Saket Saurabh. Fpt approximations for packing and covering problems parameterized by elimination distance and even less, 2023. arXiv:2310.03469.

14 Bart M. P. Jansen and Jari J. H. de Kroon. FPT algorithms to compute the elimination distance to bipartite graphs and more. In Lukasz Kowalik, Michal Pilipczuk, and Pawel Rzazewski, editors, *Graph-Theoretic Concepts in Computer Science – 47th International Workshop, WG 2021, Warsaw, Poland, June 23-25, 2021, Revised Selected Papers*, volume 12911 of *Lecture Notes in Computer Science*, pages 80–93. Springer, 2021. doi:10.1007/978-3-030-86838-3_6.

15 Bart M. P. Jansen, Jari J. H. de Kroon, and Michal Wlodarczyk. Vertex deletion parameterized by elimination distance and even less. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1757–1769. ACM, 2021. doi:10.1145/3406325.3451068.

16 Bart M. P. Jansen, Jari J. H. de Kroon, and Michal Wlodarczyk. 5-approximation for $\mathcal{H}$-treewidth essentially as fast as $\mathcal{H}$-deletion parameterized by solution size. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 66:1–66:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ESA.2023.66.

17 Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. A framework for approximation schemes on disk graphs. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2228–2241. SIAM, 2023. doi:10.1137/1.9781611977554.ch84.

18 Dániel Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008. doi:10.1093/comjnl/bxm048.

19 Laure Morelle, Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. Faster parameterized algorithms for modification problems to minor-closed classes. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 93:1–93:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.93.

20 Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009.