Polylogarithmic Approximation Algorithms for Weighted-$\mathcal{F}$-Deletion Problems

Akanksha Agrawal
Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary
agrawal.akanksha@sztaki.mta.hu

Daniel Lokshtanov
University of Bergen, Norway
daniello@ii.uib.no

Pranabendu Misra
University of Bergen, Norway
pranabendu.misra@ii.uib.no

Saket Saurabh
Institute of Mathematical Sciences, HBNI, Chennai, India,
University of Bergen, Norway, and UMI ReLax
saket@imsc.res.in

Meirav Zehavi
Ben-Gurion University, Beersheba, Israel
meiravze@bgu.ac.il

Abstract

Let $\mathcal{F}$ be a family of graphs. A canonical vertex deletion problem corresponding to $\mathcal{F}$ is defined as follows: given an n-vertex undirected graph G and a weight function $w : V(G) \to \mathbb{R}^+$, find a minimum weight subset $S \subseteq V(G)$ such that $G - S$ belongs to $\mathcal{F}$. This is known as WEIGHTED $\mathcal{F}$ VERTEX DELETION problem. In this paper we devise a recursive scheme to obtain $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-approximation algorithms for such problems, building upon the classical technique of finding *balanced separators* in a graph. Roughly speaking, our scheme applies to those problems, where an optimum solution S together with a well-structured set X, form a balanced separator of the input graph. In this paper, we obtain the first $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-approximation algorithms for the following vertex deletion problems.

- Let $\mathcal{F}$ be a finite set of graphs containing a planar graph, and $\mathcal{F} = \mathcal{G}(\mathcal{F})$ be the family of graphs such that every graph $H \in \mathcal{G}(\mathcal{F})$ excludes all graphs in $\mathcal{F}$ as minors. The vertex deletion problem corresponding to $\mathcal{F} = \mathcal{G}(\mathcal{F})$ is the WEIGHTED PLANAR $\mathcal{F}$-MINOR-FREE DELETION (WP$\mathcal{F}$-MFD) problem. We give randomized and deterministic approximation algorithms for WP$\mathcal{F}$-MFD with ratios $\mathcal{O}(\log^{1.5} n)$ and $\mathcal{O}(\log^2 n)$, respectively. Previously, only a randomized constant factor approximation algorithm for the *unweighted* version of the problem was known [FOCS 2012].
- We give an $\mathcal{O}(\log^2 n)$-factor approximation algorithm for WEIGHTED CHORDAL VERTEX DELETION (WCVD), the vertex deletion problem to the family of chordal graphs. On the way to this algorithm, we also obtain a constant factor approximation algorithm for MULTICUT on chordal graphs.
- We give an $\mathcal{O}(\log^3 n)$-factor approximation algorithm for WEIGHTED DISTANCE HEREDITARY VERTEX DELETION (WDHVD), also known as WEIGHTED RANKWIDTH-1 VERTEX DELETION (WR-1VD). This is the vertex deletion problem to the family of distance hereditary graphs, or equivalently, the family of graphs of rankwidth one.

We believe that our recursive scheme can be applied to obtain $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-approximation algorithms for many other problems as well.

© Akanksha Agrawal, Daniel Losktanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi;
licensed under Creative Commons License CC-BY

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018).
Editors: Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer; Article No. 1; pp. 1:1–1:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2012 ACM Subject Classification Mathematics of computing → Approximation algorithms

Keywords and phrases Approximation Algorithms, Planar-$\mathcal{F}$-Deletion, Separator

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2018.1

Related Version A full version of the paper is available at [2], `https://arxiv.org/abs/1707.04908`.

Funding This research has received funding from the European Research Council under ERC grant no. 306992 PARAPPROX, ERC grant no. 715744 PaPaALG, and ERC grant no. 725978 SYSTEMATICGRAPH.

Acknowledgements We sincerely thank Nikhil Bansal and Seeun William Umboh for explaining their paper to us, and for several discussions on Weighted Planar $\mathcal{F}$-Minor-Free Deletion.

1 Introduction

Let $\mathcal{F}$ be a family of undirected graphs. Then a natural optimization problem is as follows.

> Weighted $\mathcal{F}$ Vertex Deletion
> **Input:** An undirected graph G and a weight function $w : V(G) \to \mathbb{R}^+$.
> **Question:** Find a minimum weight subset $S \subseteq V(G)$ such that $G - S$ belongs to $\mathcal{F}$.

The Weighted $\mathcal{F}$ Vertex Deletion problem captures a wide class of node (or vertex) deletion problems that have been studied from the 1970s. For example, when $\mathcal{F}$ is the family of independent sets, forests, bipartite graphs, planar graphs, and chordal graphs, then the corresponding vertex deletion problem corresponds to Weighted Vertex Cover, Weighted Feedback Vertex Set, Weighted Vertex Bipartization (also called Weighted Odd Cycle Transversal), Weighted Planar Vertex Deletion and Weighted Chordal Vertex Deletion, respectively. By a classic theorem of Lewis and Yannakakis [29], the decision version of the Weighted $\mathcal{F}$ Vertex Deletion problem – deciding whether there exists a set S weight at most k, such that removing S from G results in a graph with property Π – is NP-complete for every non-trivial hereditary property[1] Π.

Characterizing the graph properties, for which the corresponding vertex deletion problems can be approximated within a bounded factor in polynomial time, is a long standing open problem in approximation algorithms [43]. In spite of a long history of research, we are still far from a complete characterization. Constant factor approximation algorithms for Weighted Vertex Cover are known since 1970s [5, 32]. Lund and Yannakakis observed that the vertex deletion problem for any hereditary property with a "finite number of minimal forbidden induced subgraphs" can be approximated within a constant ratio [30]. They conjectured that for every nontrivial, hereditary property Π with an infinite forbidden set, the corresponding vertex deletion problem cannot be approximated within a constant ratio. However, it was later shown that Weighted Feedback Vertex Set, which doesn't have a finite forbidden set, admits a constant factor approximation [3, 6], thus disproving their conjecture. On the

[1] A graph property Π is simply a family of graphs closed under isomorphism, and it is called *non-trivial* if there exists an infinite number of graphs that are in Π, as well as an infinite number of graphs that are not in Π. A non-trivial graph property Π is called *hereditary* if $G \in \Pi$ implies that every induced subgraph of G is also in Π.

other hand a result by Yannakakis [42] shows that, for a wide range of graph properties Π, approximating the minimum number of vertices to delete in order to obtain a *connected* graph with the property Π within a factor $n^{1-\varepsilon}$ is NP-hard. We refer to [42] for the precise list of graph properties to which this result applies to, but it is worth mentioning the list includes the class of acyclic graphs and the class of outerplanar graphs.

In this paper, we explore the approximability of Weighted $\mathcal{F}$ Vertex Deletion for several different families $\mathcal{F}$ and design $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-factor approximation algorithms for these problems. More precisely, our results are as follows.

1. Let $\mathcal{F}$ be a finite set of graphs that includes a planar graph. Let $\mathcal{F} = \mathcal{G}(\mathcal{F})$ be the family of graphs such that every graph $H \in \mathcal{G}(\mathcal{F})$ does not contain a graph from $\mathcal{F}$ as a minor. The vertex deletion problem corresponding to $\mathcal{F} = \mathcal{G}(\mathcal{F})$ is known as the Weighted Planar $\mathcal{F}$-Minor-Free Deletion (WP$\mathcal{F}$-MFD). The WP$\mathcal{F}$-MFD problem is a very generic problem and by selecting different sets of forbidden minors $\mathcal{F}$, one can obtain various fundamental problems such as Weighted Vertex Cover, Weighted Feedback Vertex Set or Weighted Treewidth η-Deletion. Our first result is a randomized $\mathcal{O}(\log^{1.5} n)$-factor (deterministic $\mathcal{O}(\log^2 n)$-factor) approximation algorithm for WP$\mathcal{F}$-MFD, for any finite $\mathcal{F}$ that contains a planar graph.
2. We give an $\mathcal{O}(\log^2 n)$-factor approximation algorithm for Weighted Chordal Vertex Deletion (WCVD), the vertex deletion problem corresponding to the family of chordal graphs. On the way to this algorithm, we also obtain a constant factor approximation algorithm for Weighted Multicut in chordal graphs.
3. We give an $\mathcal{O}(\log^3 n)$-factor approximation algorithm for Weighted Distance Hereditary Vertex Deletion (WDHVD), also known as the Weighted Rankwidth-1 Vertex Deletion (WR-1VD) problem. It is the vertex deletion problem corresponding to the family of distance hereditary graphs, or equivalently graphs of rankwidth 1.

All our algorithms follow the same recursive scheme: find "well structured balanced separators" in the graph by exploiting the properties of the family $\mathcal{F}$, and then use structure of the balanced separator to obtain a approximate solution. In the following, we first describe the methodology by which we design all these approximation algorithms. Then, we give a brief overview, consisting of known results and our contributions, for each problem we study. Let us also mention that these problems inherit the hardness of approximation of Vertex Cover via simple reductions. In particular, they don't admit a PTAS (polynomial time approximation scheme) unless P = NP.

Our Methods

Multicommodity max-flow min-cut theorems are a classical technique in designing approximation algorithms, which was pioneered by Leighton and Rao in their seminal paper [28]. This approach can be viewed as using balanced vertex (or edge) separators[2] in a graph to obtain a divide-and-conquer approximation algorithm. In a typical application, the optimum solution S, forms a balanced separator of the graph. Thus, the idea is to find a minimum cost balanced separator W of the graph and add it to the solution, and then recursively solve the problem on each of the connected components. This leads to an $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-factor approximation algorithm for the problem in question.

[2] Roughly speaking, a *balanced vertex separator* is a set of vertices W, such that any connected component of $G - W$ contains at most $\frac{2}{3}$ of the vertices of G.

Our recursive scheme is a strengthening of this approach which exploits the structural properties of the family $\mathcal{F}$. Here the optimum solution S^* need not be a balanced separator of the graph. Indeed, a balanced separator of the graph could be much larger than S^*. Rather, S^* along with a possibly large but well-structured subset of vertices X, forms a balanced separator of the graph. We then exploit the presence of such a balanced separator in the graph to compute an approximate solution. Consider a family $\mathcal{F}$ for which Weighted $\mathcal{F}$ Vertex Deletion is amenable to our approach, and let G be an instance of this problem. Let S be the approximate solution that we will compute. Our approximation algorithm has the following steps:

1. Find a well-structured set X, such that $G - X$ has a balanced separator W which is not too costly.
2. Next, compute the balanced separator W of $G - X$ using the known factor $\mathcal{O}(\sqrt{\log n})$-approximation algorithm (or deterministic $\mathcal{O}(\log n)$-approximation algorithm) for Weighted Vertex Separators [12, 28]. Then add W into the solution set S and recursively solve the problem on each connected component of $G-(X \cup S)$. Let $S_1, \cdots, S_\ell$ be the solutions returned by the recursive calls. We add $S_1, \cdots, S_\ell$ to the solution S.
3. Finally, we add X back into the graph and consider the instance $(G - S) \cup X$. Observe that, $V(G - S)$ can be partitioned into $V' \uplus X$, where $G[V']$ belongs to $\mathcal{F}$ and X is a well-structured set. We call such instances, the *special case* of Weighted $\mathcal{F}$ Vertex Deletion. We apply an approximation algorithm that exploits the structural properties of the special case to compute a solution.

Now consider the problem of finding the structure X. One way is to enumerate all the candidates for X and then pick the one where $G - X$ has a balanced vertex separator of least cost – this separator plays the role of W. However, the number of candidates for X in a graph could be too many to enumerate in polynomial time. For example, in the case of Weighted Chordal Vertex Deletion, the set X will be a clique in the graph, and the number of maximal cliques in a graph on n vertices could be as many as $3^{\frac{n}{3}}$ [31]. Hence, we cannot enumerate and test every candidate structure in polynomial time. However, we can exploit certain structural properties of family $\mathcal{F}$, to reduce the number of candidates for X in the graph. In our problems, we "tidy up" the graph by removing "short obstructions" that forbid the graph from belonging to the family $\mathcal{F}$. Then one can obtain an upper bound on the number of candidate structures. In the above example, recall that a graph G is chordal if and only if there are no induced cycles of length 4 or more. It is known that a graph G without any induced cycle of length 4 has at most $\mathcal{O}(n^2)$ maximal cliques [11]. Observe that, we can greedily compute a set of vertices which intersects all induced cycles of length 4 in the graph. Therefore, at the cost of factor 4 in the approximation ratio, we can ensure that the graph has only polynomially many maximal cliques. Hence, one can enumerate all maximal cliques in the remaining graph [41] to test for X.

Next consider the task of solving an instance of the special case of the problem. We again apply a recursive scheme, but now with the advantage of a much more structured graph. By a careful modification of an LP solution to the instance, we eventually reduce it to instances of Weighted Multicut. In the above example, for Weighted Chordal Vertex Deletion we obtain instances of Weighted Multicut on a chordal graph. We follow this approach for all three problems that we study in this paper. We believe our recursive scheme can be applied to obtain $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-approximation algorithms for Weighted $\mathcal{F}$ Vertex (Edge) Deletion corresponding to several other graph families $\mathcal{F}$.

Weighted Planar $\mathscr{F}$-Minor-Free Deletion. Let $\mathscr{F}$ be a finite set of graphs containing a planar graph. The vertex deletion problem corresponding to $\mathscr{F}$ is defined as follows.

Weighted Planar $\mathcal{F}$-Minor-Free Deletion (WP$\mathcal{F}$-MFD)
Input: An undirected graph G and a weight function $w : V(G) \to \mathbb{R}^+$.
Question: Find a minimum weight subset $S \subseteq V(G)$ such that $G - S$ does not contain any graph in $\mathcal{F}$ as a minor.

The WP$\mathcal{F}$-MFD problem is a very generic problem that encompasses several known problems. To explain the versatility of the problem, we require a few definitions. A graph H is called a *minor* of a graph G if we can obtain H from G by a sequence of vertex deletions, edge deletions and edge contractions, and a family of graphs $\mathcal{F}$ is called *minor closed* if $G \in \mathcal{F}$ implies that every minor of G is also in $\mathcal{F}$. Given a graph family $\mathcal{F}$, by $\mathsf{ForbidMinor}(\mathcal{F})$ we denote the family of graphs such that $G \in \mathcal{F}$ if and only if G does not contain any graph in $\mathsf{ForbidMinor}(\mathcal{F})$ as a minor. By the celebrated Graph Minor Theorem of Robertson and Seymour, every minor closed family $\mathcal{F}$ is characterized by a finite family of forbidden minors [39]. That is, $\mathsf{ForbidMinor}(\mathcal{F})$ has finite size. Indeed, the size of $\mathsf{ForbidMinor}(\mathcal{F})$ depends on the family $\mathcal{F}$. Now for a finite collection of graphs $\mathcal{F}$, as above, we may define the Weighted $\mathcal{F}$-Minor-Free Deletion problem. And observe that, even though the definition of Weighted $\mathcal{F}$-Minor-Free Deletion we only consider finite sized $\mathcal{F}$, this problem actually encompasses deletion to every minor closed family of graphs. Let $\mathcal{G}$ be the set of all finite undirected graphs, and let $\mathcal{L}$ be the family of all finite subsets of $\mathcal{G}$. Thus, every element $\mathcal{F} \in \mathcal{L}$ is a finite set of graphs, and throughout the paper we assume that $\mathcal{F}$ is explicitly given. In this paper, we show that when $\mathcal{F} \in \mathcal{L}$ contains at least one planar graph, then it is possible to obtain an $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-factor approximation algorithm for WP$\mathcal{F}$-MFD.

The case where $\mathcal{F}$ contains a planar graph, while being considerably more restricted than the general case, already encompasses a number of the well-studied instances of WP$\mathcal{F}$-MFD. For example, when $\mathcal{F} = \{K_2\}$, a complete graph on two vertices, this is the Weighted Vertex Cover problem. When $\mathcal{F} = \{C_3\}$, a cycle on three vertices, this is the Weighted Feedback Vertex Set problem. Another fundamental problem, which is also a special case of WP$\mathcal{F}$-MFD, is Weighted Treewidth-η Vertex Deletion or Weighted η-Transversal. Here the task is to delete a minimum weight vertex subset to obtain a graph of treewidth at most η. Since any graph of treewidth η excludes a $(\eta + 1) \times (\eta + 1)$ grid as a minor, we have that the set $\mathcal{F}$ of forbidden minors of treewidth η graphs contains a planar graph. Treewidth-η Vertex Deletion plays an important role in generic efficient polynomial time approximation schemes based on Bidimensionality theory [16, 17]. Other examples of Planar $\mathcal{F}$-Minor-Free Deletion problems that can be found in the literature on approximation and parameterized algorithms, are the cases of $\mathcal{F}$ being $\{K_{2,3}, K_4\}$, $\{K_4\}$, $\{\theta_c\}$, and $\{K_3, T_2\}$, which correspond to removing vertices to obtain an outerplanar graph, a series-parallel graph, a diamond graph, and a graph of pathwidth 1, respectively.

Apart from the case of Weighted Vertex Cover [5, 32] and Weighted Feedback Vertex Set [3, 6], there was not much progress on approximability/non-approximability of WP$\mathcal{F}$-MFD until the work of Fiorini, Joret, and Pietropaoli [13], which gave a constant factor approximation algorithm for the case of WP$\mathcal{F}$-MFD where $\mathcal{F}$ is a diamond graph, i.e., a graph with two vertices and three parallel edges. In 2011, Fomin et al. [14] considered Planar $\mathcal{F}$-Minor-Free Deletion (i.e. the unweighted version of WP$\mathcal{F}$-MFD) in full generality and designed a randomized (deterministic) $\mathcal{O}(\log^{1.5} n)$-factor ($\mathcal{O}(\log^2 n)$-factor) approximation algorithm for it. Later, Fomin et al. [15] gave a randomized constant factor approximation algorithm for Planar $\mathcal{F}$-Minor-Free Deletion. Our algorithm for WP$\mathcal{F}$-MFD extends this result to the weighted setting, at the cost of increasing the approximation factor to $\log^{\mathcal{O}(1)} n$.

▶ **Theorem 1.** *For every set $\mathcal{F} \in \mathcal{L}$,* WP$\mathcal{F}$-MFD *admits a randomized (deterministic) $\mathcal{O}(\log^{1.5} n)$-factor ($\mathcal{O}(\log^2 n)$-factor) approximation algorithm.*

We mention some recent related works. Bansal et al. [4] have studied the edge deletion version of the Treewidth-η Vertex Deletion problem, under the name Bounded Treewidth Interdiction Problem, and gave a bicriteria approximation algorithm. In particular, for a graph G and an integer $\eta > 0$, they gave a polynomial time algorithm that finds a subset of edges F' of G such that $|F'| \leq \mathcal{O}((\log n \log\log n) \cdot \mathsf{opt})$ and the treewidth of $G - F'$ is $\mathcal{O}(\eta \log \eta)$. In our setting where η is a fixed constant, this immediately implies a factor $\mathcal{O}(\log n \log\log n)$ approximation algorithm for the edge deletion version of WP$\mathcal{F}$-MFD.[3] However, it is not immediately clear if their approach can be extended to WP$\mathcal{F}$-MFD.[4] Very recently, Gupta et al. [22] have given $\mathcal{O}(\log \ell)$ approximation algorithm for (unweighted) Planar $\mathcal{F}$-Minor-Free Deletion, where ℓ is the maximum number of vertices in any planar graph in $\mathcal{F}$.

Weighted Chordal Vertex Deletion. This problem is defined as follows.

Weighted Chordal Vertex Deletion (WCVD) **Input:** An undirected graph G and a weight function $w : V(G) \to \mathbb{R}^+$. **Question:** Find a minimum weight subset $S \subseteq V(G)$ such that $G - S$ is a chordal graph.

The class of chordal graphs is a natural class of graphs that has been extensively studied from the viewpoints of Graph Theory and Algorithm Design. Many important problems that are NP-hard on general graphs, such as Independent Set, and Graph Coloring are solvable in polynomial time once restricted to the class of chordal graphs [21]. Recall that a graph is chordal if and only if it does not have any induced cycle of length 4 or more. Thus, Chordal Vertex Deletion (CVD) can be viewed as a natural variant of the classic Feedback Vertex Set (FVS). Indeed, while the objective of FVS is to eliminate all cycles, the CVD problem only asks us to eliminate induced cycles of length 4 or more. Despite the apparent similarity between the objectives of these two problems, the design of approximation algorithms for WCVD is very challenging. In particular, chordal graphs can be dense – indeed, a clique is a chordal graph. As we cannot rely on the sparsity of output, our approach must deviate from those employed by approximation algorithms from FVS. That being said, chordal graphs still retain some properties that resemble those of trees, and these properties are utilized by our algorithm. Prior to our work, only two non-trivial approximation algorithms for CVD were known. The first one, by Jansen and Pilipczuk [26], is a deterministic $\mathcal{O}(\mathsf{opt}^2 \log \mathsf{opt} \log n)$-factor approximation algorithm, and the second one, by Agrawal et al. [1], is a deterministic $\mathcal{O}(\mathsf{opt} \log^2 n)$-factor approximation algorithm. The second result implies that CVD admits an $\mathcal{O}(\sqrt{n} \log n)$-factor approximation algorithm.[5] In this paper we obtain the first $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-approximation algorithm for WCVD.

▶ **Theorem 2.** CVD *admits a deterministic $\mathcal{O}(\log^2 n)$-factor approximation algorithm.*

[3] One can run their algorithm first and remove the solution output by their algorithm to obtain a graph of treewidth at most $\mathcal{O}(\eta \log \eta)$. Then one can find an optimal solution using standard dynamic programming.

[4] We thank Nikhil Bansal and Seeun William Umboh for several discussions and for pointing us that their algorithm does not work for WP$\mathcal{F}$-MFD.

[5] If $\mathsf{opt} \geq \sqrt{n}/\log n$, we output a greedy solution to the input graph, and otherwise we have that $\mathsf{opt} \log^2 n \leq \sqrt{n} \log n$, hence we call the $\mathcal{O}(\mathsf{opt} \log^2 n)$-factor approximation algorithm.

While this approximation algorithm follows our general scheme, it also requires us to incorporate several new ideas. In particular, to implement the third step of the scheme, we need to design a different $\mathcal{O}(\log n)$-factor approximation algorithm for the special case of WCVD where the vertex-set of the input graph G can be partitioned into two sets, X and $V(G) \setminus X$, such that $G[X]$ is a clique and $G[V(G) \setminus X]$ is a chordal graph. This approximation algorithm is again based on recursion, but it is more involved. At each recursive call, it carefully manipulates a fractional solution of a special form. Moreover, to ensure that its current problem instance is divided into two subinstances that are independent and simpler than their origin, we introduce multicut constraints. In addition to these constraints, we keep track of the complexity of the subinstances, which is measured via the cardinality of the maximum independent set in the graph. Our multicut constraints result in an instance of Weighted Multicut, which we ensure is on a chordal graph.

> Weighted Multicut
> **Input:** An undirected graph G, a weight function $w : V(G) \to \mathbb{R}^+$ and a set $T = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of k pairs of vertices of G.
> **Question:** Find a minimum weight subset $S \subseteq V(G)$ such that for any pair $(s_i, t_i) \in \mathcal{T}$, $G - S$ does not have any path between s_i and t_i.

For Weighted Multicut on chordal graphs, no constant-factor approximation algorithm was previously known. We remark that Weighted Multicut is NP-hard on trees [19], and hence it is also NP-hard on chordal graphs. We design the first such algorithm, which our main algorithm employs as a black box.

▶ **Theorem 3.** Weighted Multicut *admits a constant-factor approximation algorithm on chordal graphs.*

This algorithm is inspired by the work of Garg, Vazirani and Yannakakis on Weighted Multicut on trees [19]. Here, we carefully exploit the well-known characterization of the class of chordal graphs as the class of graphs that admit clique forests. We believe that this result is of independent interest. The algorithm by Garg, Vazirani and Yannakakis [19] is a classic primal-dual algorithm. A more recent algorithm, by Golovin, Nagarajan and Singh [20], uses total unimodularity to obtain a different algorithm for Multicut on trees.

Weighted Distance Hereditary Vertex Deletion. Let us start with the formal definition.

> Weighted Distance Hereditary Vertex Deletion (WDHVD)
> **Input:** An undirected graph G and a weight function $w : V(G) \to \mathbb{R}^+$.
> **Question:** Find a minimum weight subset $S \subseteq V(G)$ such that $G - S$ is a distance hereditary graph.

A graph G is a *distance hereditary graph* (also called a completely separable graph [23]) if the distances between vertices in every connected induced subgraph of G are the same as in the graph G. Distance hereditary graphs were named and first studied by Hworka [25]. However, an equivalent family of graphs was earlier studied by Olaru and Sachs [40] and shown to be perfect. It was later discovered that these graphs are precisely the graphs of rankwidth one [33].

Rankwidth is a graph parameter introduced by Oum and Seymour [36] to approximate yet another graph parameter called Cliquewidth. The notion of cliquewidth was defined by Courcelle and Olariu [9] as a measure of how "clique-like" the input graph is. This is similar to the notion of treewidth, which measures how "tree-like" the input graph is. One of the main motivations was that several NP-complete problems become tractable on the family of cliques (complete graphs), the assumption was that these algorithmic properties extend to "clique-like" graphs [8]. However, computing cliquewidth and the corresponding cliquewidth decomposition seems to be computationally intractable. This then motivated the notion of rankwidth, which is a graph parameter that approximates cliquewidth well while also being algorithmically tractable [36, 34]. For more information on cliquewidth and rankwidth, we refer to the surveys by Hlinený et al. [24] and Oum [35].

As algorithms for Treewidth-η Vertex Deletion are applied as subroutines to solve many graph problems, we believe that algorithms for Weighted Rankwidth-η Vertex Deletion (WR-ηVD) will be useful in this respect. In particular, Treewidth-η Vertex Deletion has been considered in designing efficient approximation, kernelization and fixed parameter tractable algorithms for WP$\mathcal{F}$-MFD and its unweighted counterpart Planar $\mathcal{F}$-Minor-Free Deletion [4, 14, 16, 17, 18]. Along similar lines, we believe that WR-ηVD and its unweighted counterpart will be useful in designing efficient approximation, kernelization and fixed parameter tractable algorithms for Weighted $\mathcal{F}$ Vertex Deletion where $\mathcal{F}$ is characterized by a finite family of forbidden *vertex minors* [33].

Recently, Kim and Kwon [27] designed an $\mathcal{O}(\mathsf{opt}^2 \log n)$-factor approximation algorithm for Distance Hereditary Vertex Deletion (DHVD). This result implies that DHVD admits an $\mathcal{O}(n^{2/3} \log n)$-factor approximation algorithm. In this paper, we take first step towards obtaining a good approximation algorithm for WR-ηVD by designing a $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-factor approximation algorithm for WDHVD.

▶ **Theorem 4.** WDHVD *or* WR-1VD *admits an* $\mathcal{O}(\log^3 n)$*-factor approximation algorithm.*

We note that several steps of our approximation algorithm for WR-1VD can be generalized for an approximation algorithm for WR-ηVD and thus we believe that our approach should yield an $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-factor approximation algorithm for WR-ηVD. We leave that as an interesting open problem for the future.

Organization of the paper

Due to space constraints, we only present the details of Weighted Planar $\mathcal{F}$-Minor-Free Deletion in this extended abstract. The details of the algorithms for Weighted Chordal Vertex Deletion and Weighted Distance Hereditary Vertex Deletion will appear in the full version of the paper (see [2]). Graph theoretic preliminaries have been deferred to the appendix.

2 Approximation Algorithm for WP$\mathcal{F}$-MFD

In this section we prove Theorem 1. We can assume that the weight $w(v)$ of each vertex $v \in V(G)$ is positive, else we can insert v into any solution. Below we state a result from [37], which will be useful in our algorithm.

▶ **Proposition 5** ([37])**.** *Let $\mathcal{F}$ be a finite set of graphs such that $\mathcal{F}$ contains a planar graph. Then, any graph G that excludes any graph from $\mathcal{F}$ as a minor satisfies* $\mathsf{tw}(G) \leq c = c(\mathcal{F})$*.*

We let $c = c(\mathcal{F})$ to be the constant returned by Proposition 5. The approximation algorithm for WP$\mathcal{F}$-MFD comprises of two components. The first component handles the special case where the vertex set of input graph G can be partitioned into two sets C and X such that $|C| \leq c + 1$ and $H = G[X]$ is an $\mathcal{F}$-minor free graph. We note that there can be edges between vertices in C and vertices in H. We show that for these special instances, in polynomial time we can compute the size of the optimum solution and a set realizing it.

The second component is a recursive algorithm that solves general instances of the problem. Here, we gradually disintegrate the general instance until it becomes an instance of the special type where we can resolve it in polynomial time. More precisely, for each guess of $c+1$ sized subgraph M of G, we find a small separator S (using an approximation algorithm) that together with M breaks the input graph into two graphs significantly smaller than their origin. It first removes $M \cup S$, and solves each of the two resulting subinstances by calling itself recursively; then, it inserts M back into the graph, and uses the solutions it obtained from the recursive calls to construct an instance of the special case which is then solved by the first component.

2.1 Constant sized graph + $\mathcal{F}$-minor free graph

We first handle the special case where the input graph G consists of a graph M of size at most $c+1$ and an $\mathcal{F}$-minor free graph H. We refer to this algorithm as Special-WP. More precisely, along with the input graph G and the weight function w, we are also given a graph M with at most $c+1$ vertices and an $\mathcal{F}$-minor free graph H such that $V(G) = V(M) \cup V(H)$, where the vertex-sets $V(M)$ and $V(H)$ are disjoint. Note that the edge-set $E(G)$ may contain edges between vertices in M and vertices in H. We will show that such instances may be solved optimally in polynomial time. We start with the following easy observation.

▶ **Observation 6.** *Let G be a graph with $V(G) = X \uplus Y$, such that $|X| \leq c+1$ and $G[Y]$ is an $\mathcal{F}$-minor free graph. Then, the treewidth of G is at most $2c+1$.*

▶ **Lemma 7.** *Let G be a graph of treewidth t with a non-negative weight function w on the vertices, and let $\mathcal{F}$ be a finite family of graphs. Then, one can compute a minimum weight vertex set S such that $G - S$ is $\mathcal{F}$-minor free, in time $f(q,t) \cdot n$, where n is the number of vertices in G and q is a constant that depends only on $\mathcal{F}$.*

Proof. This follows from the fact that finding such a set S is expressible as an MSO-optimization formula ϕ whose length, q, depends only on the family $\mathcal{F}$ [15]. Then, by Theorem 7 [7], we can compute an optimal sized set S in time $f(q,t) \cdot n$. ◀

Now, we apply the above lemma to the graph G and the family $\mathcal{F}$, and obtain a minimum weight set S such that $G - S$ is $\mathcal{F}$-minor free.

2.2 General Graphs

We proceed to handle general instances by developing a $d \cdot \log^2 n$-factor approximation algorithm for WP$\mathcal{F}$-MFD, Gen-WP-APPROX, thus proving the correctness of Theorem 1. The exact value of the constant d will be determined later.

Recursion. We define each call to our algorithm Gen-WP-APPROX to be of the form (G', w'), where (G', w') is an instance of WP$\mathcal{F}$-MFD such that G' is an induced subgraph of G, and we denote $n' = |V(G')|$.

Goal. For each recursive call Gen-WP-APPROX(G', w'), we aim to prove the following.

▶ **Lemma 8.** Gen-WP-APPROX *returns a solution that is at least* opt *and at most $\frac{d}{2} \cdot \log^2 n' \cdot$* opt. *Moreover, it returns a subset $U \subseteq V(G')$ that realizes the solution.*

At each recursive call, the size of the graph G' becomes smaller. Thus, when we prove that Lemma 8 is true for the current call, we assume that the approximation factor is bounded by $\frac{d}{2} \cdot \log^2 \widehat{n} \cdot \mathsf{opt}$ for any call where the size $\widehat{n}$ of the vertex-set of its graph is strictly smaller than n'.

Termination. In polynomial time we can test whether G' has a minor $F \in \mathcal{F}$ [38]. Furthermore, for each $M \subseteq V(G)$ of size at most $c+1$, we can check if $G - M$ has a minor $F \in \mathcal{F}$. If $G - M$ is $\mathcal{F}$-minor free then we are in a special instance, where $G - M$ is $\mathcal{F}$ minor free and M is a constant sized graph. We optimally resolve this instance in polynomial time using the algorithm Special-WP. Since we output an optimal sized solution in the base cases, we thus ensure that at the base case of our induction Lemma 8 holds.

Recursive Call. For the analysis of a recursive call, let S^* denote a hypothetical set that realizes the optimal solution opt of the current instance (G', w'). Let (F, β) be a forest decomposition of $G' - S^*$ of width at most c, whose existence is guaranteed by Proposition 5. Using standard arguments on forests we have the following observation.

▶ **Observation 9.** *There exists a node $v \in V(F)$ such that $\beta(v)$ is a balanced separator for $G' - S^*$.*

From Observation 9 we know that there exists a node $v \in V(F)$ such that $\beta(v)$ is a balanced separator for $G' - S^*$. This together with the fact that $G' - S^*$ has treewidth at most c results in the following observation.

▶ **Observation 10.** *There exist a subset $M \subseteq V(G')$ of size at most $c+1$ and a subset $S \subseteq V(G') \setminus M$ of weight at most opt such that $M \cup S$ is a balanced separator for G'.*

This gives us a polynomial time algorithm as stated in the following lemma.

▶ **Lemma 11.** *There is a deterministic (randomized) algorithm which in polynomial-time finds $M \subseteq V(G')$ of size at most $c+1$ and a subset $S \subseteq V(G') \setminus M$ of weight at most $q \cdot \log n' \cdot \mathsf{opt}$ ($q^* \cdot \sqrt{\log n'} \cdot \mathsf{opt}$) for some fixed constant q (q^*) such that $M \cup S$ is a balanced separator for G'.*

Proof. Note that we can enumerate every $M \subseteq V(G')$ of size at most $c+1$ in time $\mathcal{O}(n^c)$. For each such M, we can either run the randomized $q^* \cdot \sqrt{\log n'}$-factor approximation algorithm by Feige et al. [12] or the deterministic $q \cdot \log n'$-factor approximation algorithm by Leighton and Rao [28] to find a balanced separator S_M of $G' - M$. Here, q and q^* are fixed constants. By Observation 10, there is a set S in $\{S_M : M \subseteq V(G')$ and $M \leq c+1\}$ such that $w(S) \leq q \cdot \log n' \cdot \mathsf{opt}$ ($w(S) \leq q^* \cdot \sqrt{\log n'} \cdot \mathsf{opt}$). Thus, the desired output is a pair (M, S) where M is one of the vertex subset of size at most $c+1$ such that $S_M = S$. ◀

We call the algorithm in Lemma 11 to obtain a pair (M, S). Since $M \cup S$ is a balanced separator for G', we can partition the set of connected components of $G' - (M \cup S)$ into two sets, $\mathcal{A}_1$ and $\mathcal{A}_2$, such that for $V_1 = \bigcup_{A \in \mathcal{A}_1} V(A)$ and $V_2 = \bigcup_{A \in \mathcal{A}_2} V(A)$ it holds that $n_1, n_2 \leq \frac{2}{3} n'$ where $n_1 = |V_1|$ and $n_2 = |V_2|$. We remark that we use different algorithms for finding a balanced separator in Lemma 11 based on whether we are looking for a randomized algorithm or a deterministic algorithm.

Next, we define two inputs of (the general case of) WP$\mathcal{F}$-MFD: $I_1 = (G'[V_1], w'|_{V_1})$ and $I_2 = (G'[V_2], w'|_{V_2})$. Let opt_1 and opt_2 denote the optimal solutions to I_1 and I_2, respectively. Observe that since $V_1 \cap V_2 = \emptyset$, it holds that $\mathsf{opt}_1 + \mathsf{opt}_2 \leq \mathsf{opt}$. We solve each of the subinstances by recursively calling algorithm Gen-WP-APPROX. By the inductive hypothesis, we thus obtain two sets, S_1 and S_2, such that $G'[V_1] - S_1$ and $G'[V_2] - S_2$ are $\mathcal{F}$-minor free graphs, and $w'(S_1) \leq \frac{d}{2} \cdot \log^2 n_1 \cdot \mathsf{opt}_1$ and $w'(S_2) \leq \frac{d}{2} \cdot \log^2 n_2 \cdot \mathsf{opt}_2$.

We proceed by defining an input of the special case of WP$\mathcal{F}$-MFD: $J = (G'[(V_1 \cup V_2 \cup M) \setminus (S_1 \cup S_2)], w'|_{(V_1 \cup V_2 \cup M) \setminus (S_1 \cup S_2)})$. Observe that $G'[V_1 \setminus S_1]$ and $G'[V_2 \setminus S_2]$ are $\mathcal{F}$-minor free graphs and there are no edges between vertices in V_1 and vertices in V_2 in $G' - M$, and M is of constant size. Therefore, we resolve this instance by calling algorithm Special-WP. We thus obtain a set, $\widehat{S}$, such that $G'[(V_1 \cup V_2 \cup M) \setminus (S_1 \cup S_2 \cup \widehat{S})]$ is a $\mathcal{F}$-minor graph, and $w'(\widehat{S}) \leq \mathsf{opt}$ (since $|(V_1 \cup V_2 \cup M) \setminus (S_1 \cup S_2)| \leq n'$ and the optimal solution of each of the special subinstances is at most opt).

Observe that any obstruction in $G' - S$ is either completely contained in $G'[V_1]$, or completely contained in $G'[V_2]$, or it contains at least one vertex from M. This observation, along with the fact that $G'[(V_1 \cup V_2 \cup M) \setminus (S_1 \cup S_2 \cup \widehat{S})]$ is a $\mathcal{F}$-minor free graph, implies that $G' - T$ is a $\mathcal{F}$-minor free graph where $T = S \cup S_1 \cup S_2 \cup \widehat{S}$. Thus, it is now sufficient to show that $w'(T) \leq \frac{d}{2} \cdot (\log n')^2 \cdot \mathsf{opt}$.

By the discussion above, we have that

$$\begin{aligned} w'(T) &\leq w'(S) + w'(S_1) + w'(S_2) + w'(\widehat{S}) \\ &\leq q \cdot \log n' \cdot \mathsf{opt} + \tfrac{d}{2} \cdot ((\log n_1)^2 \cdot \mathsf{opt}_1 + (\log n_2)^2 \cdot \mathsf{opt}_2) + \mathsf{opt} \end{aligned}$$

Recall that $n_1, n_2 \leq \frac{2}{3} n'$ and $\mathsf{opt}_1 + \mathsf{opt}_2 \leq \mathsf{opt}$. Thus, we have that

$$\begin{aligned} w'(T) &< q \cdot \log n' \cdot \mathsf{opt} + \tfrac{d}{2} \cdot (\log \tfrac{2}{3} n')^2 \cdot \mathsf{opt} + \mathsf{opt} \\ &< \tfrac{d}{2} \cdot (\log n')^2 \cdot \mathsf{opt} + \log n' \cdot \mathsf{opt} \cdot (q + 1 + \tfrac{d}{2} \cdot (\log \tfrac{3}{2})^2 - \tfrac{d}{2} \cdot 2 \cdot \log \tfrac{3}{2}). \end{aligned}$$

Overall, we conclude that to ensure that $w'(T) \leq \frac{d}{2} \cdot \log^2 n' \cdot \mathsf{opt}$, it is sufficient to ensure that $q + 1 + \frac{d}{2} \cdot (\log \frac{3}{2})^2 - \frac{d}{2} \cdot 2 \cdot \log \frac{3}{2} \leq 0$, which can be done by fixing $d = \dfrac{2}{2 \log \frac{3}{2} - (\log \frac{3}{2})^2} \cdot (q + 1)$.

If we use the $\mathcal{O}(\sqrt{\log n})$-factor approximation algorithm by Feige et al. [12] for finding a balance separator in Lemma 11, then we can do the analysis similar to the deterministic case and obtain a randomized factor-$\mathcal{O}(\log^{1.5} n)$approximation algorithm for WP$\mathcal{F}$-MFD.

3 Conclusion

In this paper, we designed $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-approximation algorithms for Weighted Planar $\mathcal{F}$-Minor-Free Deletion, Weighted Chordal Vertex Deletion and Weighted Distance Hereditary Vertex Deletion (or Weighted Rankwidth-1 Vertex Deletion). These algorithms are the first ones for these problems whose approximation factors are bounded by $\mathcal{O}(\log^{\mathcal{O}(1)} n)$. Along the way, we also obtained a constant-factor approximation algorithm for Weighted Multicut on chordal graphs. All our algorithms are based on the same recursive scheme. We believe that the scope of applicability of our approach is very wide. We would like to conclude our paper with the following concrete open problems.

- Does Weighted Planar $\mathcal{F}$-Minor-Free Deletion admit a constant-factor approximation algorithm? Furthermore, studying families $\mathcal{F}$ that do not necessarily contain a planar graph is another direction for further research.
- Does Weighted Chordal Vertex Deletion admit a constant-factor approximation algorithm?
- Does Weighted Rankwidth-η Vertex Deletion admit a $\mathcal{O}(\log^{\mathcal{O}(1)} n)$-factor approximation algorithm?
- On which other graph classes Weighted Multicut admits a constant-factor approximation?

References

1 Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Feedback vertex set inspired kernel for chordal vertex deletion. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1383–1398, 2017.

2 Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Polylogarithmic approximation algorithms for weighted-$\mathcal{F}$-deletion problems. *arXiv preprint arXiv:1707.04908*, 2017.

3 Vineet Bafna, Piotr Berman, and Toshihiro Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM Journal on Discrete Mathematics*, 12(3):289–297, 1999.

4 Nikhil Bansal, Daniel Reichman, and Seeun William Umboh. LP-based robust algorithms for noisy minor-free and bounded treewidth graphs. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1964–1979, 2017.

5 Reuven Bar-Yehuda and Shimon Even. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2(2):198–203, 1981.

6 Reuven Bar-Yehuda, Dan Geiger, Joseph Naor, and Ron M. Roth. Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and Bayesian inference. *SIAM Journal on Computing*, 27(4):942–959, 1998.

7 Richard B. Borie, R. Gary Parker, and Craig A. Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7(5&6):555–581, 1992.

8 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.

9 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

10 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

11 M Farber. On diameters and radii of bridged graphs. *Discrete Mathematics*, 73:249–260, 1989.

12 Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM Journal on Computing*, 38(2):629–657, 2008.

13 Samuel Fiorini, Gwenaël Joret, and Ugo Pietropaoli. Hitting diamonds and growing cacti. In *Proceedings of the 14th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 6080, pages 191–204, 2010.

14 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Hitting forbidden minors: Approximation and kernelization. *SIAM Journal on Discrete Mathematics*, 30(1):383–410, 2016.

15 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar f-deletion: Approximation, kernelization and optimal fpt algorithms. In *Proceedings of IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 470–479, 2012.

16 Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Bidimensionality and EPTAS. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 748–759, 2011.

17 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In *Proceedings of the 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1563–1575, 2012.

18 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. In *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 503–510, 2010.

19 Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.

20 Daniel Golovin, Viswanath Nagarajan, and Mohit Singh. Approximating the k-multicut problem. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 621–630, 2006.

21 Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.

22 Anupam Gupta, Euiwoong Lee, Jason Li, Pasin Manurangsi, and Michał Włodarczyk. Losing treewidth by separating subsets. *arXiv preprint arXiv:1804.01366*, 2018.

23 Peter L Hammer and Frédéric Maffray. Completely separable graphs. *Discrete applied mathematics*, 27(1):85–99, 1990.

24 Petr Hlinený, Sang-il Oum, Detlef Seese, and Georg Gottlob. Width parameters beyond tree-width and their applications. *The Computer Journal*, 51(3):326–362, 2008.

25 Edward Howorka. A characterization of distance-hereditary graphs. *The quarterly journal of mathematics*, 28(4):417–420, 1977.

26 Bart M. P. Jansen and Marcin Pilipczuk. Approximation and kernelization for chordal vertex deletion. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1399–1418, 2017.

27 E. J. Kim and O. Kwon. A Polynomial Kernel for Distance-Hereditary Vertex Deletion. *ArXiv e-prints*, 2016. `arXiv:1610.07229`.

28 T Leighton and S Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46:787—-832, 1999.

29 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

30 Carsten Lund and Mihalis Yannakakis. The approximation of maximum subgraph problems. In *Proceedings of the 20nd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 700, pages 40–51, 1993.

31 John W Moon and Leo Moser. On cliques in graphs. *Israel journal of Mathematics*, 3(1):23–28, 1965.

32 G. L. Nemhauser and L. E. Trotter, Jr. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6:48–61, 1974.

33 Sang-il Oum. Rank-width and vertex-minors. *Journal of Combinatorial Theory, Series B*, 95(1):79–100, 2005.

34 Sang-il Oum. Approximating rank-width and clique-width quickly. *ACM Transactions on Algorithms*, 5(1), 2008.

35 Sang-il Oum. Rank-width: Algorithmic and structural results. *CoRR*, abs/1601.03800, 2016.

36 Sang-il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006.

37 Neil Robertson and P D Seymour. Graph minors. v. excluding a planar graph. *Journal of Combinatorial Theory Series B*, 41(1):92–114, 1986.

38 Neil Robertson and Paul D. Seymour. Graph minors .xiii. the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.

39 Neil Robertson and Paul D. Seymour. Graph minors. XX. wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

40 Horst Sachs. On the berge conjecture concerning perfect graphs. *Combinatorial Structures and their Applications*, 37:384, 1970.

41 Shuji Tsukiyama, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.

42 Mihalis Yannakakis. The effect of a connectivity requirement on the complexity of maximum subgraph problems. *Journal of the ACM*, 26(4):618–630, 1979.

43 Mihalis Yannakakis. Some open problems in approximation. In *Proceedings of 2nd Italian Conference on Algorithms and Complexity, Second (CIAC)*, pages 33–39, 1994.

A Preliminaries

For a positive integer k, we use $[k]$ as a shorthand for $\{1, 2, \ldots, k\}$. Given a function $f : A \to B$ and a subset $A' \subseteq A$, we let $f|_{A'}$ denote the function f restricted to the domain A'.

Graphs. Given a graph G, we let $V(G)$ and $E(G)$ denote its vertex-set and edge-set, respectively. In this paper, we only consider undirected graphs. We let $n = |V(G)|$ denote the number of vertices in the graph G, where G will be clear from context. The *open neighborhood*, or simply the *neighborhood*, of a vertex $v \in V(G)$ is defined as $N_G(v) = \{w \mid \{v, w\} \in E(G)\}$. The *closed neighborhood* of v is defined as $N_G[v] = N_G(v) \cup \{v\}$. The *degree* of v is defined as $d_G(v) = |N_G(v)|$. We can extend the definition of the neighborhood of a vertex to a set of vertices as follows. Given a subset $U \subseteq V(G)$, $N_G(U) = \bigcup_{u\in U} N_G(u)$ and $N_G[U] = \bigcup_{u\in U} N_G[u]$. The *induced subgraph* $G[U]$ is the graph with vertex-set U and edge-set $\{\{u, u'\} \mid u, u' \in U,$ and $\{u, u'\} \in E(G)\}$. Moreover, we define $G - U$ as the induced subgraph $G[V(G) \setminus U]$. We omit subscripts when the graph G is clear from context. For graphs G and H, by $G \cap H$, we denote the graph with vertex set as $V(G) \cap V(H)$ and edge set as $E(G) \cap E(H)$. An *independent set* in G is a set of vertices such that there is no edge in G between any pair of vertices in this set. The *independence number* of G, denoted by $\alpha(G)$, is defined as the cardinality of the largest independent set in G. A *clique* in G is a set of vertices such that there is an edge in G between every pair of vertices in this set.

A *path* $P = (x_1, x_2, \ldots, x_\ell)$ in G is a subgraph of G where $V(P) = \{x_1, x_2, \ldots, x_\ell\} \subseteq V(G)$ and $E(P) = \{\{x_1, x_2\}, \{x_2, x_3\}, \ldots, \{x_{\ell-1}, x_\ell\}\} \subseteq E(G)$, where $\ell \in [n]$. The vertices x_1 and x_ℓ are called the *endpoints* of the path P and the remaining vertices in $V(P)$ are called the *internal vertices* of P. We also say that P is a path between x_1 and x_ℓ or connects x_1 and x_ℓ. A *cycle* $C = (x_1, x_2, \ldots, x_\ell)$ in G is a subgraph of G where $V(C) = \{x_1, x_2, \ldots, x_\ell\} \subseteq V(G)$ and $E(C) = \{\{x_1, x_2\}, \{x_2, x_3\}, \ldots, \{x_{\ell-1}, x_\ell\}, \{x_\ell, x_1\}\} \subseteq E(G)$, i.e., it is a path with an additional edge between x_1 and x_ℓ. The graph G is *connected* if there is a path between every pair of vertices in G, otherwise G is *disconnected*. A connected graph without any cycles is a *tree*, and a collection of trees is a *forest*. A maximal connected subgraph of G is called a *connected component* of G. Given a function $f : V(G) \to \mathbb{R}$ and a subset $U \subseteq V(G)$, we denote $f(U) = \sum_{v\in U} f(v)$. Moreover, we say that a subset $U \subseteq V(G)$ is a *balanced separator for* G if for each connected component C in $G - U$, it holds that $|V(C)| \leq \frac{2}{3}|V(G)|$. We refer the reader to [10] for details on standard graph theoretic notations and terminologies that are not explicitly defined here.

Forest Decompositions. A *forest decomposition* of a graph G is a pair (F, β) where F is forest, and $\beta : V(T) \to 2^{V(G)}$ is a function that satisfies the following:

1. $\bigcup_{v\in V(F)} \beta(v) = V(G)$;
2. for any edge $\{v, u\} \in E(G)$, there is a node $w \in V(F)$ such that $v, u \in \beta(w)$;
3. for any $v \in V(G)$, the collection of nodes $T_v = \{u \in V(F) \mid v \in \beta(u)\}$ is a subtree of F.

For $v \in V(F)$, we call $\beta(v)$ the *bag* of v, and for the sake of clarity of presentation, we sometimes use v and $\beta(v)$ interchangeably. We refer to the vertices in $V(F)$ as *nodes*. A *tree decomposition* is a forest decomposition where F is a tree. For a graph G, by $\mathsf{tw}(G)$ we denote the minimum over all possible *tree decompositions* of G, the maximum size of a bag minus one in that *tree decomposition*.

Minors. Given a graph G and an edge $\{u, v\} \in E(G)$, the graph G/e denotes the graph obtained from G by contracting the edge $\{u, v\}$, that is, the vertices u, v are deleted from G and a new vertex $uv^\star$ is added to G which is adjacent to the all the neighbors of u, v previously in G (except for u, v). A graph H that is obtained by a sequence of edge contractions in G is said to be a contraction of G. A graph H is a *minor* of a G if H is the contraction of some subgraph of G. We say that a graph G is F-minor free when F is not a minor of G. Given a family $\mathcal{F}$ of graphs, we say that a graph G is $\mathcal{F}$-minor free, if for all $F \in \mathcal{F}$, F is not a minor of G. It is well known that if H is a minor of G, then $\mathsf{tw}(H) \leq \mathsf{tw}(G)$. A graph is *planar* if it is $\{K_5, K_{3,3}\}$-minor free [10]. Here, K_5 is a clique on 5 vertices and $K_{3,3}$ is a complete bipartite graph with both sides of bipartition having 3 vertices.

Chordal Graphs. Let G be a graph. For a cycle C on at least four vertices, we say that $\{u, v\} \in E(G)$ is a *chord* of C if $u, v \in V(C)$ but $\{u, v\} \notin E(C)$. A cycle C is *chordless* if it contains at least four vertices and has no chords. The graph G is a *chordal graph* if it has no chordless cycle as an induced subgraph. A *clique forest* of G is a forest decomposition of G where every bag is a maximal clique. The following lemma shows that the class of chordal graphs is exactly the class of graphs which have a clique forest.

▶ **Lemma 12** ([21]). *A graph G is a chordal graph if and only if G has a clique forest. Moreover, a clique forest of a chordal graph can be constructed in polynomial time.*

Given a subset $U \subseteq V(G)$, we say that U *intersects* a chordless cycle C in G if $U \cap V(C) \neq \emptyset$. Observe that if U *intersects* every chordless cycle of G, then $G - U$ is a chordal graph.