

Hitting forbidden minors: Approximation and Kernelization

Fedor V. Fomin¹, Daniel Lokshtanov², Neeldhara Misra³,
Geevarghese Philip³, and Saket Saurabh³

- 1 Department of Informatics, University of Bergen, N-5020 Bergen, Norway.
fomin@ii.uib.no
- 2 University of California, San Diego, Department of Computer Science and
Engineering, 9500 Gilman Drive, La Jolla, CA 92093-0404, USA,
dlokshtanov@cs.ucsd.edu
- 3 The Institute of Mathematical Sciences, Chennai 600113, India.
{neeldhara|gphilip|saket}@imsc.res.in

Abstract

We study a general class of problems called p - \mathcal{F} -DELETION problems. In an p - \mathcal{F} -DELETION problem, we are asked whether a subset of at most k vertices can be deleted from a graph G such that the resulting graph does not contain as a minor any graph from the family \mathcal{F} of forbidden minors. We obtain a number of algorithmic results on the p - \mathcal{F} -DELETION problem when \mathcal{F} contains a planar graph. We give

- a linear vertex kernel on graphs excluding t -claw $K_{1,t}$, the star with t leaves, as an induced subgraph, where t is a fixed integer.
- an approximation algorithm achieving an approximation ratio of $O(\log^{3/2} OPT)$, where OPT is the size of an optimal solution on general undirected graphs.

Finally, we obtain polynomial kernels for the case when \mathcal{F} only contains graph θ_c as a minor for a fixed integer c . The graph θ_c consists of two vertices connected by c parallel edges. Even though this may appear to be a very restricted class of problems it already encompasses well-studied problems such as VERTEX COVER, FEEDBACK VERTEX SET and DIAMOND HITTING SET. The generic kernelization algorithm is based on a non-trivial application of protrusion techniques, previously used only for problems on topological graph classes.

Digital Object Identifier 10.4230/LIPIcs.STACS.2011.189

1 Introduction

Let \mathcal{F} be a finite set of graphs. In an p - \mathcal{F} -DELETION problem, we are given an n -vertex graph G and an integer k as input, and asked whether at most k vertices can be deleted from G such that the resulting graph does not contain a graph from \mathcal{F} as a minor. We refer to such a subset as an \mathcal{F} -hitting set. The p - \mathcal{F} -DELETION problem is a generalization of several fundamental problems. For example, when $\mathcal{F} = \{K_2\}$, a complete graph on two vertices, this is the VERTEX COVER problem. When $\mathcal{F} = \{C_3\}$, a cycle on three vertices, this is the FEEDBACK VERTEX SET problem. Other famous cases are $\mathcal{F} = \{K_{2,3}, K_4\}$, $\mathcal{F} = \{K_{3,3}, K_5\}$ and $\mathcal{F} = \{K_3, T_2\}$, which correspond to removing vertices to obtain outerplanar graphs, planar graphs, and graphs of pathwidth one respectively. Here, $K_{i,j}$ denotes the complete bipartite graph with bipartitions of sizes i and j , and K_i denotes the complete graph on i vertices. Further, a T_2 is a star on three leaves, each of whose edges has been subdivided exactly once. A T_2 structure is depicted in the leftmost graph of Figure 1.

Our interest in the p - \mathcal{F} -DELETION problem is motivated by its generality and the recent developments in kernelization or polynomial time preprocessing. The parameterized



© F. V. Fomin, D. Lokshtanov, N. Misra, G. Philip and S. Saurabh;
licensed under Creative Commons License NC-ND

28th Symposium on Theoretical Aspects of Computer Science (STACS'11).

Editors: Thomas Schwentick, Christoph Dürr; pp. 189–200

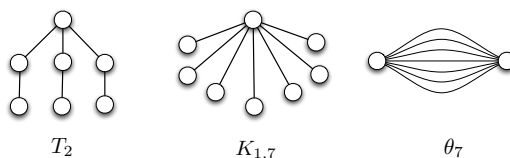
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



SYMPOSIUM
ON THEORETICAL
ASPECTS
OF COMPUTER
SCIENCE



■ **Figure 1** Graphs T_2 , t -claw $K_{1,t}$ with $t = 7$, and θ_c with $c = 7$

complexity of this general problem is well understood. By a celebrated result of Robertson and Seymour, every p - \mathcal{F} -DELETION problem is non-uniformly fixed-parameter tractable (FPT). That is, for every k there is an algorithm solving the problem in time $O(f(k) \cdot n^3)$ [32]. In this paper we study this problem from the view point of polynomial time preprocessing and approximation, when the obstruction set \mathcal{F} satisfies certain properties.

Preprocessing as a strategy for coping with hard problems is universally applied in practice and the notion of *kernelization* provides a mathematical framework for analyzing the quality of preprocessing strategies. We consider parameterized problems, where every instance I comes with a *parameter* k . Such a problem is said to admit a *polynomial kernel* if every instance (I, k) can be reduced in polynomial time to an equivalent instance with both size and parameter value bounded by a polynomial in k . The study of kernelization is a major research frontier of Parameterized Complexity and many important recent advances in the area are on kernelization. These include general results showing that certain classes of parameterized problems have polynomial kernels [3, 10, 22, 26]. The recent development of a framework for ruling out polynomial kernels under certain complexity-theoretic assumptions [9, 17, 23] has added a new dimension to the field and strengthened its connections to classical complexity. For overviews of kernelization we refer to surveys [8, 24] and to the corresponding chapters in books on Parameterized Complexity [21, 30].

While the initial interest in kernelization was driven mainly by practical applications, the notion of kernelization turned out to be very important in theory as well. It is well known, see e.g. [18], that a parameterized problem is fixed parameter tractable, or belongs to the class FPT, if and only if it has a (perhaps exponential) kernel. Kernelization enables us to classify problems within the class FPT further, based on the sizes of the problem kernels. So far, most of the work done in the field of kernelization is still specific to particular problems and powerful unified techniques to identify classes of problems with polynomial kernels are still in their nascent stage. One of the fundamental challenges in the area is the possibility of characterising general classes of parameterized problems possessing kernels of polynomial sizes. From this perspective, the class of the p - \mathcal{F} -DELETION problems is very interesting because it contains as special cases the p -VERTEX COVER and p -FEEDBACK VERTEX SET problems which are the most intensively studied problems from the kernelization perspective.

Our contribution and key ideas. One of the main conceptual contributions of this work is the extension of protrusion techniques, initially developed in [10, 22] for obtaining meta-kernelization theorems for problems on sparse graphs like planar and H -minor-free graphs, to general graphs. We demonstrate this by obtaining a number of kernelization results for the p - \mathcal{F} -DELETION problem when \mathcal{F} contains a planar graph. Our first result is the following theorem for graphs that do not contain $K_{1,t}$ (a star on t leaves, see Figure 1).

► **Theorem 1.** *Let \mathcal{F} be an obstruction set containing a planar graph. Then p - \mathcal{F} -DELETION admits a linear vertex kernel on graphs excluding $K_{1,t}$ as an induced subgraph, where t is a fixed integer.*

Several well studied graph classes do not contain graphs with induced $K_{1,t}$. Of course, every graph with maximum vertex degree at most $(t-1)$ is $K_{1,t}$ -free. The class of $K_{1,3}$ -free graphs, also known as claw-free graphs, contains line graphs and de Bruijn graphs. Unit disc graphs are known to be $K_{1,7}$ -free [15].

Our kernelization is a divide and conquer algorithm which finds large protrusions. A *protrusion* is a subgraph of constant treewidth separated from the remaining part of the graph by a constant number of vertices. Having found protrusions of substantial size, the kernelization algorithm replaces them with smaller, “equivalent” protrusions. Here we use the results from the work by Bodlaender et al. [10] that enable this step whenever the parameterized problem in question “behaves like a regular language”. To prove that p - \mathcal{F} -DELETION has the desired properties for this step, we formulate the problem in monadic second order logic and show that it exhibits certain monotonicity properties. As a corollary we obtain that p -FEEDBACK VERTEX SET, p -DIAMOND HITTING SET, p -PATHWIDTH ONE DELETION SET, and p -OUTERPLANAR DELETION SET admit a linear vertex kernel on graphs excluding $K_{1,t}$ as an induced subgraph. With the same methodology we also obtain a $O(k \log k)$ vertex kernel for p -DISJOINT CYCLE PACKING on graphs excluding $K_{1,t}$ as an induced subgraph. We note that p -DISJOINT CYCLE PACKING does not admit a polynomial kernel on general graphs [11] unless $coNP \subseteq NP/poly$.

Let θ_c be a graph with two vertices and $c \geq 1$ parallel edges (see Figure 1). Our second result is the following theorem on general graphs.

► **Theorem 2.** *Let \mathcal{F} be an obstruction set containing only θ_c . Then p - \mathcal{F} -DELETION admits a kernel of size $O(k^2 \log^{3/2} k)$.*

A number of well-studied NP-hard combinatorial problems are special cases of p - θ_c -DELETION. When $c = 1$, this is the classical VERTEX COVER problem [29]. For $c = 2$, this is another well studied problem, the FEEDBACK VERTEX SET problem [4]. When $c = 3$, this is the DIAMOND HITTING SET problem [20]. Let us note that the size of the best known kernel for $c = 2$ is $O(k^2)$, which is very close to the size of the kernel in Theorem 2. Also, Dell and van Melkebeek proved that no NP-hard vertex deletion problem based on a graph property that is inherited by subgraphs can have kernels of size $O(k^{2-\epsilon})$ unless $coNP \subseteq NP/poly$ [17] and thus the sizes of the kernels in Theorem 2 are tight up to a polylogarithmic factor.

The proof of Theorem 2 is obtained in a series of non-trivial steps. The very high level idea is to reduce the general case to problem on graphs of bounded degree, which allows us to use the protrusion techniques as in the proof of Theorem 1. However, vertex degree reduction is not straightforward and requires several new ideas. One of the new tools is a generic $O(\log^{3/2} OPT)$ -approximation algorithm for the p - \mathcal{F} -DELETION problem when the class of excluded minors for \mathcal{F} contains at least one planar graph. More precisely, we obtain the following result, which is of independent interest.

► **Theorem 3.** *Let \mathcal{F} be an obstruction set containing a planar graph, and let OPT be the size of the smallest \mathcal{F} -hitting set. Given a graph G , in polynomial time we can find a subset $S \subseteq V(G)$ such that $G[V \setminus S]$ contains no element of \mathcal{F} as a minor and $|S| = O(OPT \cdot \log^{3/2} OPT)$.*

While several generic approximation algorithms are known for problems of minimum vertex deletion to obtain subgraphs with property P , like when P is a hereditary property with a finite number of minimal forbidden subgraphs [28], or can be expressed as a universal first order sentence over subsets of edges of the graph [25], we are not aware of any generic approximation algorithm for the case when a property P is characterized by a finite set of forbidden minors.

We then use the approximation algorithm as a subroutine in a polynomial time algorithm that transforms the input instance (G, k) into an equivalent instance (G', k') such that $k' \leq k$ and the maximum degree of G' is bounded by $O(k \log^{3/2} k)$. An important combinatorial tool used in designing this algorithm is the q -Expansion Lemma. For $q = 1$ this lemma is Hall's theorem and its usage is equivalent to the application of the Crown Decomposition technique [1, 14]. After obtaining an equivalent instance with bounded degree, we apply protrusion techniques and prove Theorem 2.

Related work. All non-trivial p - \mathcal{F} -DELETION problems are NP-hard [27]. By one of the most well-known consequences of the celebrated Graph Minor theory of Robertson and Seymour the p - \mathcal{F} -DELETION problem is non-uniformly fixed parameter tractable. Whenever \mathcal{F} is given explicitly, the problem is uniformly FPT because the excluded minors for the class of graphs that are YES-instances of the p - \mathcal{F} -DELETION problem can be computed explicitly [2]. A special case of that problem, when the set \mathcal{F} contains θ_c , has been studied from approximation and parameterized perspectives. In particular, the case of p - θ_1 -DELETION or, equivalently, p -VERTEX COVER, is the most well-studied problem in Parameterized Complexity. Different kernelization techniques were applied on the problem, eventually resulting in a $2k$ -sized vertex kernel [1, 13]. For the kernelization of p -FEEDBACK VERTEX SET, or p - θ_2 -DELETION, there has been a sequence of dramatic improvements starting from an $O(k^{11})$ vertex kernel by Buragge et al. [12], improved to $O(k^3)$ by Bodlaender [7], and then finally to $O(k^2)$ by Thomassé [34]. Recently Philip et al. [31] and Cygan et al. [16] obtained polynomial kernels for p -PATHWIDTH ONE DELETION SET. Constant factor approximation algorithms are known for VERTEX COVER and FEEDBACK VERTEX SET [4, 5]. Very recently, a constant factor approximation algorithm for the DIAMOND HITTING SET problem, or p - θ_3 -DELETION, was obtained in [20]. Prior to our work, no polynomial kernels were known for p -DIAMOND HITTING SET or more general families of p - \mathcal{F} -DELETION problems.

2 Preliminaries

In this section we give various definitions which we use in the paper. For $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \dots, n\}$. We use $V(G)$ to denote the vertex set of a graph G , and $E(G)$ to denote the edge set. The degree of a vertex v in G is the number of edges incident on v , and is denoted by $d(v)$. We use $\Delta(G)$ to denote the maximum degree of G . A graph G' is a *subgraph* of G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. The subgraph G' is called an *induced subgraph* of G if $E(G') = \{\{u, v\} \in E(G) \mid u, v \in V(G')\}$. Given a subset $S \subseteq V(G)$ the subgraph induced by S is denoted by $G[S]$. The subgraph induced by $V(G) \setminus S$ is denoted by $G \setminus S$. We denote by $N(S)$ the open neighborhood of S , i.e. the set of vertices in $V(G) \setminus S$ adjacent to S . Let \mathcal{F} be a finite set of graphs. A vertex subset $S \subseteq V(G)$ of a graph G is said to be a \mathcal{F} -*hitting set* if $G \setminus S$ does not contain any graphs in the family \mathcal{F} as a minor.

By *contracting* an edge (u, v) of a graph G , we mean identifying the vertices u and v , keeping all the parallel edges and removing all the loops. A *minor* of a graph G is a graph H that can be obtained from a subgraph of G by contracting edges. We keep parallel edges after contraction since the graph θ_c which we want to exclude as a minor itself contains parallel edges. Let G, H be two graphs. A subgraph G' of G is said to be a *minor-model* of H in G if G' contains H as a minor. The subgraph G' is a *minimal minor-model* of H in G if no proper subgraph of G' is a minor-model of H in G . A graph class \mathcal{C} is *minor closed* if any minor of any graph in \mathcal{C} is also an element of \mathcal{C} . A minor closed graph class \mathcal{C} is *H -minor-free* or simply *H -free* if $H \notin \mathcal{C}$.

Parameterized algorithms and Kernels. A parameterized problem Π is a subset of

$\Gamma^* \times \mathbb{N}$ for some finite alphabet Γ . An instance of a parameterized problem consists of (x, k) , where k is called the parameter. A central notion in parameterized complexity is *fixed parameter tractability (FPT)* which means, for a given instance (x, k) , solvability in time $f(k) \cdot p(|x|)$, where f is an arbitrary function of k and p is a polynomial in the input size. A kernelization algorithm for a parameterized problem $\Pi \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, outputs, in time polynomial in $(|x| + k)$, a pair $(x', k') \in \Sigma^* \times \mathbb{N}$ such that: (a) $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$ and (b) $|x'|, k' \leq g(k)$, where g is some computable function. The output instance x' is called the kernel, and the function g is referred to as the size of the kernel. If $g(k) = k^{O(1)}$, then we say that Π admits a polynomial kernel [21].

Tree-width and protrusions. We use standard notions of tree decompositions, tree-width, and nice tree decompositions, which definitions can be found in [10, 22]. Whenever we use nice tree decompositions, we will assume that the root bag is empty. The tree-width of a graph G is denoted by $tw(G)$. Given a graph G and $S \subseteq V(G)$, we define $\partial_G(S)$ as the set of vertices in S that have a neighbor in $V(G) \setminus S$. We say that a set $X \subseteq V(G)$ is an r -protrusion of G if $tw(G[X]) \leq r$ and $|\partial_G(X)| \leq r$. In our paper, we also use concepts of MSO, t -boundaried graphs and their properties, the notion of finite integer index, and strong monotonicity. The definitions of these notions can be found in [10, 22]. Proofs of results labeled with \star have been omitted due to lack of space.

3 Kernelization for p - \mathcal{F} -Deletion on $K_{1,t}$ free graphs

In this section we show that if the obstruction set \mathcal{F} contains a planar graph then the p - \mathcal{F} -DELETION problem has a linear vertex kernel on graphs excluding $K_{1,t}$ as an induced subgraph. We start with the following lemma, which is crucial to our kernelization algorithms.

► **Lemma 4 (\star).** *Let \mathcal{F} be an obstruction set containing a planar graph of size h . If G has an \mathcal{F} -hitting set S of size at most k , then $tw(G \setminus S) \leq d$ and $tw(G) \leq k + d$, where $d = 20^{2(14h-24)^5}$.*

The Protrusion Rule — Reductions Based on Finite Integer Index. We obtain our kernelization algorithm for p - \mathcal{F} -DELETION by applying a protrusion based reduction rule. That is, any large r -protrusion for a fixed constant r that depends only on \mathcal{F} (that is, only on the problem) is replaced with a smaller equivalent r -protrusion. For this, we utilize the following lemma of Bodlaender et al. [10].

► **Lemma 5 ([10]).** *Let Π be a problem that has finite integer index. Then there exists a computable function $\gamma : \mathbb{N} \rightarrow \mathbb{N}$ and an algorithm that given an instance (G, k) and an r -protrusion X of G of size at least $\gamma(r)$, runs in $O(|X|)$ time and outputs an instance (G^*, k^*) such that $|V(G^*)| < |V(G)|$, $k^* \leq k$, and $(G^*, k^*) \in \Pi$ if and only if $(G, k) \in \Pi$.*

Remark: Let us remark that if G does not have $K_{1,t}$ as an induced subgraph then the proof of Lemma 5 also ensures that the graph G' does not contain $K_{1,t}$ as an induced subgraph. This ensures that the reduced instance belongs to the same graph class as the original.

In order to apply Lemma 5 we need to be able to efficiently find large r -protrusions whenever the instance considered is large enough. Also, we need to prove that p - \mathcal{F} -DELETION has finite integer index. The next lemma yields a divide and conquer algorithm for efficiently finding large r -protrusions.

► **Lemma 6.** *There is a linear time algorithm that given an n -vertex graph G and a set $X \subseteq V(G)$ such that $tw(G \setminus X) \leq d$, outputs a $2(d+1)$ -protrusion of G of size at least $\frac{n-|X|}{4^{|N(X)|+1}}$. Here d is some constant.*

Proof. Let $F = G \setminus X$. The algorithm starts by computing a nice tree decomposition of F of width at most d . Notice that since d is a constant this can be done in linear time [6]. Let S be the vertices in $V(F)$ that are neighbors of X in G , that is, $S = N_G(X)$.

The nice tree decomposition of F is a pair $(T, \mathcal{B} = \{B_\ell\}_{\ell \in V(T)})$, where T is a rooted binary tree. We will now *mark* some of the nodes of T . For every $v \in S$, we mark the topmost node ℓ in T such that $v \in B_\ell$. In this manner, at most $|S|$ nodes are marked. Now we mark more nodes of T by exhaustively applying the following rule: if u and v are marked, mark their least common ancestor in T . Let M be the set of all marked nodes of T . Standard counting arguments on trees give that $|M| \leq 2|S|$.

Since T is a binary tree, it follows that $T \setminus M$ has at most $2|M| + 1$ connected components. Let the vertex sets of these connected components be $C_1, C_2 \dots C_\eta$, $\eta \leq 2|M| + 1$. For every $i \leq \eta$, let $C'_i = N_T(C_i) \cup C_i$ and let $P_i = \bigcup_{u \in C'_i} B_u$. By the construction of M , every component of $T \setminus M$ has at most 2 neighbors in M . Also for every $1 \leq i \leq \eta$ and $v \in S$, we have that if $v \in P_i$, then v should be contained in one of the bags of $N_T(C_i)$. In other words, $S \cap P_i \subseteq \bigcup_{u \in C'_i \setminus C_i} B_u$. Thus every P_i is a $2(d+1)$ -protrusion of G . Since $\eta \leq 2|M| + 1 \leq 4|S| + 1$, the pigeon-hole principle yields that there is a protrusion P_i with at least $\frac{n-|X|}{4|S|+1}$ vertices. The algorithm constructs M and $P_1 \dots P_\eta$ and outputs the largest protrusion P_i . It is easy to implement this procedure to run in linear time. \blacktriangleleft

The following lemma follows from [10], where it is shown that every strongly monotone p -MIN-MSO problem has finite integer index.

► **Lemma 7 (★).** p - \mathcal{F} -DELETION has finite integer index.

Analysis and Kernel Size – Proof of Theorem 1. Now we give the desired kernel for p - \mathcal{F} -DELETION. We will use the following combinatorial lemma.

► **Lemma 8 (★).** Let G be a graph excluding $K_{1,t}$ as an induced subgraph and S be an \mathcal{F} -hitting set. If \mathcal{F} contains a planar graph of size h , then $|N(S)| \leq g(h, t) \cdot |S|$ for some computable function g of h and t .

Proof of Theorem 1. Let (G, k) be an instance of p - \mathcal{F} -DELETION and h be the size of a smallest planar graph in the obstruction set \mathcal{F} . We first apply Theorem 3 (to be proved in next section), an approximation algorithm for p - \mathcal{F} -DELETION with factor $O(\log^{3/2} OPT)$, and obtain a set X such that $G \setminus X$ contains no graph in \mathcal{F} as a minor. If the size of the set X is more than $O(k \log^{3/2} k)$ then we return that (G, k) is a NO-instance to p - \mathcal{F} -DELETION. This is justified by the approximation guarantee provided by the Theorem 3.

Let d denote the treewidth of the graph after the removal of X , that is, $d := \text{tw}(G \setminus S)$. Now we obtain the kernel in two phases: we first apply the protrusion rule selectively (Lemma 5) and get a polynomial kernel. Then, we apply the protrusion rule exhaustively on the obtained kernel to get a smaller kernel. This is done in order to reduce the running time complexity of the kernelization algorithm. To obtain the kernel we follow the following steps.

Applying the Protrusion Rule. By Lemma 4, $d \leq 20^{2(14h-24)^5}$. We apply Lemma 6 and obtain a $2(d+1)$ -protrusion Y of G of size at least $\frac{|V(G')|-|X|}{4|N(X)|+1}$. By Lemma 7, p - \mathcal{F} -DELETION has finite integer index. Let $\gamma : \mathbb{N} \rightarrow \mathbb{N}$ be the function defined in Lemma 5. If $\frac{|V(G')|-|X|}{4|N(X)|+1} \geq \gamma(2d+1)$, then using Lemma 5 we replace the $2(d+1)$ -protrusion Y in G and obtain an instance (G^*, k^*) such that $|V(G^*)| < |V(G)|$, $k^* \leq k$, and (G^*, k^*) is a YES-instance of p - \mathcal{F} -DELETION if and only if (G, k) is a YES-instance of p - \mathcal{F} -DELETION. Recall that G^* also excludes $K_{1,t}$ as an induced subgraph.

Let (G^*, k^*) be a reduced instance with hitting set X . In other words, there is no $(2d+2)$ -protrusion of size $\gamma(2d+2)$ in $G^* \setminus X$, and Protrusion Rule no longer applies. We

claim that the number of vertices in this graph is bounded by $O(k \log^{3/2} k)$. Indeed, since we cannot apply the Protrusion Rule, we have that $\frac{|V(G^*)| - |X|}{4|N(X)| + 1} \leq \gamma(2d + 2)$. Because $k^* \leq k$, we have that $|V(G^*)| \leq \gamma(2d + 2)(4|N(X)| + 1) + |X|$. By Lemma 8, $|N(X)| \leq g(h, d) \cdot |X|$ and thus $|V(G^*)| = O(\gamma(2d + 2) \cdot k \log^{3/2} k) = O(k \log^{3/2} k)$. This gives us a polynomial time algorithm that returns a vertex kernel of size $O(k \log^{3/2} k)$.

Now we give a kernel of smaller size. We would like to replace every large $(2d + 2)$ -protrusion in graph by a smaller one. We find a $(2d + 2)$ -protrusion Y of size at least $\gamma(2d + 2)$ by guessing the boundary $\partial(Y)$ of size at most $2d + 2$. This could be performed in time $k^{O(d)}$. So let (G^*, k^*) be the reduced instance on which we cannot apply the Protrusion Rule. If G is a YES-instance then there is a \mathcal{F} -hitting set X of size at most k such that $\text{tw}(G \setminus X) \leq d$. Now applying the analysis above with this X yields that $|V(G^*)| = O(k)$. Hence if the number of vertices in the reduced instance G^* , to which we can not apply the Protrusion Rule, is more than $O(k)$ then we return that G is a NO-instance. This concludes the proof of the theorem. \blacktriangleleft

► **Corollary 9.** p -FEEDBACK VERTEX SET, p -DIAMOND HITTING SET, p -PATHWIDTH ONE DELETION SET, p -OUTERPLANAR DELETION SET admit linear vertex kernel on graphs excluding $K_{1,t}$ as an induced subgraph.

The methodology used in proving Theorem 1 is not limited to p - \mathcal{F} -DELETION. For example, it is possible to obtain an $O(k \log k)$ vertex kernel on $K_{1,t}$ -free graphs for p -DISJOINT CYCLE PACKING, which is for a given graph G and positive integer k to determine if there are k vertex disjoint cycles in G . It is interesting to note that p -DISJOINT CYCLE PACKING does not admit a polynomial kernel on general graphs [11]. For our kernelization algorithm, we use the following Erdős-Pósa property [19]: given a positive integer ℓ every graph G either has ℓ vertex disjoint cycles or there exists a set $S \subseteq V(G)$ of size at most $O(\ell \log \ell)$ such that $G \setminus S$ is a forest. So given a graph G and an integer k , we first apply the factor 2 approximation algorithm given in [4] and obtain a set S such that $G \setminus S$ is a forest. If the size of S is more than $O(k \log k)$ then we return that G has k vertex disjoint cycles. Else, we use the fact that p -DISJOINT CYCLE PACKING [10] has finite integer index and apply the protrusion reduction rule in $G \setminus S$ to obtain an equivalent instance (G^*, k^*) , as in Theorem 1. The analysis for kernel size used in the proof of Theorem 1 together with the observation that $\text{tw}(G \setminus S) \leq 1$ shows that if (G, k) is a YES instance then the size of $V(G^*)$ is at most $O(k \log k)$.

► **Corollary 10.** p -DISJOINT CYCLE PACKING has $O(k \log k)$ vertex kernel on graphs excluding $K_{1,t}$ as an induced subgraph.

Next, we extend the methods used in this section for obtaining kernels for p - \mathcal{F} -DELETION on graphs excluding $K_{1,t}$ as an induced graph to all graphs, though for restricted \mathcal{F} — we consider the families \mathcal{F} that contain θ_c . However, for this kernelization result, we need a polynomial time approximation algorithm with a factor polynomial in optimum size and not depending on the input size. For example, an approximation algorithm with factor $O(\log n)$ would not serve our purpose. We obtain an approximation algorithm (Theorem 3) for p - \mathcal{F} -DELETION with a factor $O(\log^{3/2} OPT)$ whenever the finite obstruction set \mathcal{F} contains a planar graph. Here OPT is the size of a minimum \mathcal{F} -hitting set. This immediately implies a factor $O(\log^{3/2} n)$ algorithm for all the problems that can be categorized by p - \mathcal{F} -DELETION. The proof of Theorem 3 is crucially based on the following lemma.

► **Lemma 11 (★).** *There is a polynomial time algorithm that, given a graph G and a positive integer k , either reports that G has no \mathcal{F} -hitting set of size at most k or finds an \mathcal{F} -hitting set of size at most $O(k \log^{3/2} k)$.*

4 Kernelization for p - θ_c -Deletion

In this section we obtain a polynomial kernel for p - θ_c -DELETION on general graphs. To obtain our kernelization algorithm we not only need the approximation algorithm mentioned before but also a variation of classical Hall's theorem. We first present this combinatorial tool and other auxiliary results that we make use of.

Combinatorial Lemma and some Linear-Time Subroutines. We need a variation of the celebrated Hall's Theorem, which we call the q -Expansion Lemma. It is a generalization of a result due to Thomassé [34, Theorem 2.3], and captures a certain property of neighborhood sets in graphs that implicitly has been used by several authors to obtain polynomial kernels for many graph problems. For $q = 1$, the application of this lemma is exactly the well-known Crown Reduction Rule [1, 14].

The Expansion Lemma. Consider a bipartite graph G with vertex bipartition $A \uplus B$. Given subsets $S \subseteq A$ and $T \subseteq B$, we say that S has $|S|$ q -stars in T if to every $x \in S$ we can associate a subset $F_x \subseteq N(x) \cap T$ such that (a) for all $x \in S$, $|F_x| = q$; (b) for any pair of vertices $x, y \in S$, $F_x \cap F_y = \emptyset$. Observe that if S has $|S|$ q -stars in T then every vertex x in S could be thought of as the center of a star with its q leaves in T , with all these stars being vertex-disjoint. Further, a collection of $|S|$ q -stars is also a family of q edge-disjoint matchings, each saturating S . We use the following result in our kernelization algorithm to bound the degrees of vertices.

► **Lemma 12 (★).** [The q -Expansion Lemma] *Let q be a positive integer, and let m be the size of the maximum matching in a bipartite graph G with vertex bipartition $A \uplus B$. If $|B| > mq$, and there are no isolated vertices in B , then there exist nonempty vertex sets $S \subseteq A, T \subseteq B$ such that S has $|S|$ q -stars in T and no vertex in T has a neighbor outside S . Furthermore, the sets S, T can be found in time polynomial in the size of G .*

► **Observation 1 (★).** For $c \geq 2$, any minimal θ_c minor-model M of a graph G is a connected subgraph of G , and does not contain a vertex whose degree in M is less than 2, or a vertex whose deletion from M results in a disconnected graph (a cut vertex of M).

► **Lemma 13 (★).** *Let G be a graph and v a vertex of G . Given a tree decomposition of width $t \in O(1)$ of G , in $O(n)$ time we can find both (1) a smallest set $S \subseteq V$ of vertices of G such that the graph $G \setminus S$ does not contain θ_c as a minor, and (2) a largest collection $\{M_1, M_2, \dots, M_l\}$ of θ_c minor models of G such that for $1 \leq i < j \leq l$, $(V(M_i) \cap V(M_j)) = \{v\}$.*

Now we describe the reduction rules used by the kernelization algorithm. In contrast to the reduction rules employed by most known kernelization algorithms, these rules cannot always be applied on general graphs in polynomial time. Hence the algorithm does not proceed by applying these rules exhaustively, as is typical in kernelization programs. We describe how to arrive at situations where these rules can in fact be applied in polynomial time, and prove that even this selective application of rules results in a kernel of size polynomial in the parameter k .

Bounding the Maximum Degree of a Graph Now we present a set of reduction rules which, given an input instance (G, k) of p - θ_c -DELETION, obtains an equivalent instance (G', k') where $k' \leq k$ and the maximum degree of G' is at most a polynomial in k . In the sequel a vertex v is *irrelevant* if it is not a part of any θ_c minor model, and is *relevant* otherwise. For each rule below, the input instance is (G, k) .

► **Reduction Rule 1 (Irrelevant Vertex Rule).** Delete all irrelevant vertices in G .

Given a graph G and a vertex $v \in V(G)$, an ℓ -flower passing through v is a set of ℓ different θ_c minor-models in G , each containing v and no two sharing any vertex other than v .

► **Reduction Rule 2 (Flower Rule).** If a $(k+1)$ -flower passes through a vertex v of G , then include v in the solution and remove it from G to obtain the equivalent instance $(G \setminus \{v\}, (k-1))$.

The argument for the soundness of these reduction rules is simple and is hence omitted. One can test whether a particular vertex v is part of any minimal minor-model corresponding to θ_c using the rooted minor testing algorithm of Robertson and Seymour [32]. It is not clear, however, that one might check whether a vertex is a part of $(k+1)$ -flower in polynomial time. Hence we defer the application of these rules and apply them only when the vertices are “evidently” irrelevant or finding a flower can be solved in polynomial time. Now we state an auxiliary lemma which will be useful in bounding the maximum degree of the graph.

► **Lemma 14 (★).** *Let G be a n -vertex graph containing θ_c as a minor and v be a vertex such that $G' = G \setminus \{v\}$ does not contain θ_c as a minor and the maximum size of a flower containing v is at most k . Then there exists a set T_v of size $O(k)$ such that $v \notin T_v$ and $G \setminus T_v$ does not contain θ_c as a minor. Moreover we can find the set T_v in polynomial time.*

► **Lemma 15.** *There exists a polynomial time algorithm that, given an instance (G, k) of p - θ_c -DELETION returns an equivalent instance (G', k') such that $k' \leq k$ and that the maximum degree of G' is $O(k \log^{3/2} k)$. Moreover it also returns a θ_c -hitting set of G' of size $O(k \log^{3/2} k)$.*

Proof. Given an instance (G, k) of p - θ_c -DELETION, we first apply Lemma 11 on (G, k) . The polynomial time algorithm described in Lemma 11, given a graph G and a positive integer k either reports that G has no θ_c -hitting set of size at most k , or finds a θ_c -hitting set of size at most $k^* = O(k \log^{3/2} k)$. If the algorithm reports that G has no θ_c -hitting set of size at most k , then we return that (G, k) is a NO-instance to p - θ_c -DELETION. So we assume that we have a hitting set \mathcal{S} of size k^* . Now we proceed with the following two rules.

Selective Flower Rule. To apply the Flower Rule selectively we use \mathcal{S} , the θ_c -hitting set. For a vertex $v \in \mathcal{S}$ let $\mathcal{S}_v := \mathcal{S} \setminus \{v\}$ and let $G_v := G \setminus \mathcal{S}_v$. By a result of Robertson et. al. [33] we know that any graph of treewidth greater than 20^{2c^5} contains a $c \times c$ grid, and hence θ_c , as a minor. Since deleting v from G_v makes it θ_c -minor-free, $\text{tw}(G_v) \leq 20^{2c^5} + 1 = O(1)$. Now by Lemma 13, we find in linear time the size of the largest flower centered at v , in G_v . If for any vertex $v \in \mathcal{S}$ the size of the flower in G_v is at least $k+1$, we apply the Flower Rule and get an equivalent instance $(G \leftarrow G \setminus \{v\}, k \leftarrow k-1)$. Furthermore, we set $\mathcal{S} := \mathcal{S} \setminus \{v\}$. We apply the Flower Rule selectively until no longer possible. We abuse notation and continue to use (G, k) to refer to the instance that is reduced with respect to exhaustive application of the Selective Flower Rule. Thus, for every vertex $v \in \mathcal{S}$ the size of any flower passing through v in G_v is at most k .

Now we describe how to find, for a given $v \in V(G)$, a hitting set $H_v \subseteq V(G) \setminus \{v\}$ for all minor-models of θ_c that contain v . Notice that this hitting set is required to *exclude* v , so H_v cannot be the trivial hitting set $\{v\}$. If $v \notin \mathcal{S}$, then $H_v = \mathcal{S}$. On the other hand, suppose $v \in \mathcal{S}$. Since the maximum size of a flower containing v in the graph G_v is at most k , by Lemma 14, we can find a set T_v of size $O(k)$ that does not contain v and hits all the θ_c minor-models passing through v in G_v . Hence in this case we set $H_v = \mathcal{S}_v \cup T_v$. We denote $|H_v|$ by h_v . Notice that H_v is defined algorithmically, that is, there could be many small hitting sets in $V(G) \setminus \{v\}$ hitting all minor-models containing v , and H_v is one of them.

q -expansion Rule with $q = c$. Given an instance (G, k) , \mathcal{S} , and a family of sets H_v , we show that if there is a vertex v with degree more than $ch_v + c(c-1)h_v$, then we can reduce its degree to at most $ch_v + c(c-1)h_v$ by repeatedly applying the q -Expansion Lemma with $q = c$. Observe that for every vertex v the set H_v is also a θ_c hitting set for G , that is, H_v

hits *all* minor-models of θ_c in G . Consider the graph $G \setminus H_v$. Let the components of this graph that contain a neighbor of v be C_1, C_2, \dots, C_r . Note that v cannot have more than $(c - 1)$ neighbors into any component, else contracting the component will form a θ_c minor and will contradict the fact that H_v hits all the θ_c minors. Also note that none of the C_i 's can contain a minor model of θ_c .

We say that a component C_i is adjacent to H_v if there exists a vertex $u \in C_i$ and $w \in H_v$ such that $(u, w) \in E(G)$. Next we show that vertices in components that are not adjacent to H_v are irrelevant in G . Recall a vertex is irrelevant if there is no minimal minor model of θ_c that contains it. Consider a vertex u in a component C that is not adjacent to H_v . Since $G[V(C) \cup \{v\}]$ does not contain any θ_c minor we have that if u is a part of a minimal minor model $M \subseteq G$, then $v \in M$ and also there exists a vertex $u' \in M$ such that $u' \notin C \cup \{v\}$. Then the removal of v disconnects u from u' in M , a contradiction to Observation 1 that for $c \geq 2$, any minimal θ_c minor model M of a graph G does not contain a cut vertex. Applying the Irrelevant Vertex Rule to the vertices in all such components leaves us with a new set of components D_1, D_2, \dots, D_s , such that for every i , in D_i , there is at least one vertex that is adjacent to a vertex in H_v .

As before, we continue to use G to refer to the graph obtained after the Irrelevant Vertex Rule has been applied in the context described above. We also update the sets H_v for $v \in V(G)$ by deleting all the vertices w from these sets those have been removed using Irrelevant Vertex Rule.

Now, consider a bipartite graph \mathcal{G} with vertex bipartitions H_v and D . Here $D = \{d_1, \dots, d_s\}$ contains a vertex d_i corresponding to each component D_i . For every $u \in H_v$, we add the edge (u, d_i) if there is a vertex $w \in D_i$ such that $\{u, w\} \in E(G)$. Even though we start with a simple graph (graphs without parallel edges) it is possible that after applying reduction rules parallel edges may appear. However, throughout the algorithm, we ensure that the number of parallel edges between any pair of vertices is at most c . Now, v has at most ch_v edges to vertices in H_v . Since v has at most $(c - 1)$ edges to each D_i , it follows that if $d(v) > ch_v + c(c - 1)h_v$, then the number of components $|D|$ is more than ch_v . Now by applying q -Expansion Lemma with $q = c$, $A = H_v$, and $B = D$, we find a subset $S \subseteq H_v$ and $T \subseteq D$ such that S has $|S|$ c -stars in T and $N(T) = S$.

The reduction rule involves deleting edges of the form (v, u) for all $u \in D_i$, such that $d_i \in T$, and adding c edges between v and w for all $w \in S$. We add these edges only if they were not present before so that the number of edges between any pair of vertices remains at most c . This completes the description of the q -expansion reduction rule with $q = c$. Let G_R be the graph obtained after applying the reduction rule. The following lemma shows the correctness of the rule.

► **Lemma 16 (★)**. *Let G , S and v be as above and G_R be the graph obtained after applying the c -expansion rule. Then (G, k) is a YES instance of p - θ_c -DELETION if and only if (G_R, k) is a YES instance of p - θ_c -DELETION.*

Observe that all edges that are added during the application of the q -expansion reduction rule have at least one end point in \mathcal{S} , and hence \mathcal{S} remains a hitting set of G_R . We are now ready to summarize the algorithm that bounds the degree of the graph (see Algorithm 1).

Let the instance output by Algorithm 1 be (G', k', \mathcal{S}) . Clearly, in G' , the degree of every vertex is at most $ch_v + c(c - 1)h_v \leq O(k \log^{3/2} k)$. The routine also returns \mathcal{S} — a θ_c -hitting set of G' of size at most $O(k \log^{3/2} k)$.

We now show that the algorithm runs in polynomial time. For $x \in V(G)$, let $\nu(x)$ be the number of neighbors of x to which x has fewer than c parallel edges. Observe that the

Algorithm 1 BOUND-DEGREE(G, k, \mathcal{S})

-
- 1: Apply the Selective Flower Rule
 - 2: **if** $\exists v \in V(G)$ such that $d(v) > ch_v + c(c-1)h_v$ **then**
 - 3: Apply the q -expansion reduction rule with $q = c$.
 - 4: **else**
 - 5: Return (G, k, \mathcal{S}) .
 - 6: **end if**
 - 7: Return BOUND-DEGREE(G, k, \mathcal{S}).
-

application of q -expansion reduction rule never increases $\nu(x)$ for any vertex and decreases $\nu(x)$ for at least one vertex. The other rules delete vertices, which can never increase $\nu(x)$ for any vertex. This concludes the proof. ◀

5 Conclusion

In this paper we gave the first kernelization algorithms for a subset of p - \mathcal{F} -DELETION problems and a generic approximation algorithm for the p - \mathcal{F} -DELETION problem when the set of excluded minors \mathcal{F} contains at least one planar graph. Our approach generalizes and unifies known kernelization algorithms for p -VERTEX COVER and p -FEEDBACK VERTEX SET. By the celebrated result of Robertson and Seymour, every p - \mathcal{F} -DELETION problem is FPT and our work naturally leads to the following question: does every p - \mathcal{F} -DELETION problem have a polynomial kernel? Can it be that for some finite sets of minor obstructions $\mathcal{F} = \{O_1, \dots, O_p\}$ the answer to this question is NO? Even the case $\mathcal{F} = \{K_5, K_{3,3}\}$, vertex deletion to planar graphs, is an interesting challenge. Another interesting question is if our techniques can be extended to another important case when \mathcal{F} contains a planar graph.

References

- 1 F. N. Abu-Khzam, M. R. Fellows, M. A. Langston, and W. H. Suters. Crown structures for vertex cover kernelization. *Theory Comput. Syst.*, 41(3):411–430, 2007.
- 2 I. Adler, M. Grohe, and S. Kreutzer. Computing excluded minors. In *SODA*, pages 641–650, 2008.
- 3 N. Alon, G. Gutin, E. J. Kim, S. Szeider, and A. Yeo. Solving max-r-sat above a tight lower bound. In *SODA*, pages 511–517. ACM-SIAM, 2010.
- 4 V. Bafna, P. Berman, and T. Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM J. Discr. Math.*, 12(3):289–297, 1999.
- 5 R. Bar-Yehuda and S. Even. A linear-time approximation algorithm for the weighted vertex cover problem. *J. Algorithms*, 2(2):198–203, 1981.
- 6 H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.
- 7 H. L. Bodlaender. A cubic kernel for feedback vertex set. In *STACS*, volume 4393 of *Lecture Notes in Comput. Sci.*, pages 320–331. Springer, 2007.
- 8 H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *IWPEC*, volume 5917 of *Lecture Notes in Computer Science*, pages 17–37. Springer, 2009.
- 9 H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.
- 10 H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. (Meta) Kernelization. In *FOCS*, pages 629–638. IEEE, 2009.

- 11 H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel Bounds for Disjoint Cycles and Disjoint Paths. In *ESA*, volume 5757 of *Lecture Notes in Comput. Sci.*, pages 635–646. Springer, 2009.
- 12 K. Burrage, V. Estivill Castro, M. R. Fellows, M. A. Langston, S. Mac, and F. A. Rosamond. The Undirected Feedback Vertex Set Problem Has a Poly(k) Kernel. In *IWPEC*, volume 4169 of *Lecture Notes in Comput. Sci.*, pages 192–202. Springer, 2006.
- 13 J. Chen, I. A. Kanj, and W. Jia. Vertex cover: further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.
- 14 B. Chor, M. R. Fellows, and D. W. Juedes. Linear kernels in linear time, or how to save k colors in $O(n^2)$ steps. In *WG*, volume 3353 of *LNCS*, pages 257–269. Springer, 2004.
- 15 B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Math.*, 86(1-3):165–177, 1990.
- 16 M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Improved fpt algorithm and quadratic kernel for pathwidth one vertex deletion. In *IPEC*, volume 6478 of *Lecture Notes in Comput. Sci.*, pages 95–106. Springer, 2010.
- 17 H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *STOC*, pages 251–260. ACM, 2010.
- 18 R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1998.
- 19 P. Erdős and L. Pósa. On independent circuits contained in a graph. *Canadian J. Math.*, 17:347–352, 1965.
- 20 S. Fiorini, G. Joret, and U. Pietropaoli. Hitting diamonds and growing cacti. In *IPCO 2010*, volume 6080 of *Lecture Notes in Comput. Sci.*, pages 191–204. Springer, 2010.
- 21 J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.
- 22 F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. In *SODA*, pages 503–510. ACM-SIAM, 2010.
- 23 L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *STOC*, pages 133–142. ACM, 2008.
- 24 J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007.
- 25 P. G. Kolaitis and M. N. Thakur. Approximation properties of NP minimization classes. *J. Comput. System Sci.*, 50:391–411, 1995.
- 26 S. Kratsch. Polynomial kernelizations for $\text{MIN } F^+_{\text{PI}_1}$ and MAX NP . In *STACS*, volume 3 of (*LIPICs*), pages 601–612. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2009.
- 27 J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. of Comp. System Sci.*, 20(2):219 – 230, 1980.
- 28 C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. In *ICALP*, volume 700 of *Lecture Notes in Comput. Sci.*, pages 40–51. Springer, 1993.
- 29 G. L. Nemhauser and L. E. Trotter, Jr. Properties of vertex packing and independence system polyhedra. *Math. Programming*, 6:48–61, 1974.
- 30 R. Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2006.
- 31 G. Philip, V. Raman, and Y. Villanger. A quartic kernel for pathwidth-one vertex deletion. In *WG*, volume 6410 of *Lecture Notes in Computer Science*, pages 196–207, 2010.
- 32 N. Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Comb. Theory Ser. B*, 63:65–110, 1995.
- 33 N. Robertson, P. D. Seymour, and R. Thomas. Quickly excluding a planar graph. *J. Comb. Theory Ser. B*, 62:323–348, 1994.
- 34 S. Thomassé. A quadratic kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.