Uniform Kernelization Complexity of Hitting Forbidden Minors

Archontia C. Giannopoulou[1], Bart M.P. Jansen[2(✉)], Daniel Lokshtanov[3], and Saket Saurabh[4]

[1] University of Warsaw, Warsaw, Poland
archontia.giannopoulou@gmail.com
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
b.m.p.jansen@tue.nl
[3] University of Bergen, Bergen, Norway
daniello@ii.uib.no
[4] Institute of Mathematical Sciences, Chennai, India
saket@imsc.res.in

Abstract. The $\mathcal{F}$-Minor-Free Deletion problem asks, for a fixed set $\mathcal{F}$ and an input consisting of a graph G and integer k, whether k vertices can be removed from G such that the resulting graph does not contain any member of $\mathcal{F}$ as a minor. Fomin et al. (FOCS 2012) showed that the special case when $\mathcal{F}$ contains at least one planar graph has a kernel of size $f(\mathcal{F}) \cdot k^{g(\mathcal{F})}$ for some functions f and g. They left open whether this Planar $\mathcal{F}$-Minor-Free Deletion problem has kernels whose size is uniformly polynomial, of the form $f(\mathcal{F}) \cdot k^c$ for some universal constant c. We prove that some Planar $\mathcal{F}$-Minor-Free Deletion problems do not have uniformly polynomial kernels (unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$), not even when parameterized by the vertex cover number. On the positive side, we consider the problem of determining whether k vertices can be removed to obtain a graph of treedepth at most η. We prove that this problem admits uniformly polynomial kernels with $\mathcal{O}(k^6)$ vertices for every fixed η.

Keywords: Kernelization · Treedepth · Minor-free deletion

1 Introduction

Kernelization is the subfield of parameterized and multivariate algorithmics that investigates the power of provably effective preprocessing procedures for hard combinatorial problems. In kernelization we study *parameterized problems*: decision problems where every instance x is associated with a parameter k that

Supported by ERC Grant 267959 and the Warsaw Center of Mathematics and Computer Science (A.G.), NWO Veni grant “Frontiers in Parameterized Preprocessing” and NWO Gravity grant “Networks” (B.M.P.J.), Bergen Research Foundation grant BeHard (D.L.), and ERC Starting Grant “Parameterized Approximation” (S.S.).

M.M. Halldórsson et al. (Eds.): ICALP 2015, Part I, LNCS 9134, pp. 629–641, 2015.
DOI: 10.1007/978-3-662-47672-7_51

measures some aspect of its structure. A parameterized problem is said to admit a kernel of size $f\colon \mathbb{N} \to \mathbb{N}$ if every instance (x, k) can be reduced in polynomial time to an equivalent instance with both size and parameter value bounded by $f(k)$. For practical and theoretical reasons we are primarily interested in kernels whose size is polynomial, so-called *polynomial kernels*.

One of the fundamental challenges in the area is the possibility of characterizing general classes of parameterized problems possessing a kernel of polynomial size. In other words, to obtain "kernelization meta-theorems". In general, algorithmic meta-theorems have the following form: problems definable in a certain logic admit a certain kind of algorithms on certain inputs. A typical example of a meta-theorem is Courcelle's celebrated theorem which states that all graph properties definable in monadic second order logic can be decided in linear time on graphs of bounded treewidth. It seems very difficult to find a fragment of logic for which every problem expressible in this logic admits a polynomial kernel on all undirected graphs. The main obstacle in obtaining such results stems from the fact that even a simplest form of logic can formalize problems that are not even fixed parameter tractable (FPT). In graph theory, one can define a general family of problems as follows. Let $\mathcal{F}$ be a family of graphs. Given an undirected graph G and a positive integer k, is it possible to do at most k edits of G such that the resulting graph does not contain a graph from $\mathcal{F}$? Here one can define edits as either vertex/edge deletions, edge additions, or edge contraction. Similarly, one may consider containment as a subgraph, induced subgraph, or a minor. The topic of this paper is one such generic problem, namely, the $\mathcal{F}$-Minor-Free Deletion problem. It asks, for a fixed set of graphs $\mathcal{F}$ and an input consisting of a graph G and integer k, whether k vertices can be removed from G such that the resulting graph does not contain any member of $\mathcal{F}$ as a minor. The problem can also be viewed as finding a set of k vertices that hit all the minor models of $H \in \mathcal{F}$ in G, which explains the title. The parameterized complexity of this general problem is well understood: for every k there is an algorithm solving the problem in time $f(k) \cdot n^3$ [1,20]. Thus, the $\mathcal{F}$-Minor-Free Deletion problem is an interesting subject from the kernelization perspective: For which sets $\mathcal{F}$ does $\mathcal{F}$-Minor-Free Deletion admit a polynomial kernel?

Fomin et al. [11] studied the special case where $\mathcal{F}$ contains at least one planar graph, known as Planar $\mathcal{F}$-Minor-Free Deletion. It is much more restricted than $\mathcal{F}$-Minor-Free Deletion, but still generalizes problems such as Vertex Cover and Feedback Vertex Set. These problems are essentially about deleting k vertices to get a graph of constant treewidth: graphs that exclude a planar graph H as a minor have treewidth at most $|V(H)|^{\mathcal{O}(1)}$ [4]. Fomin et al. [11] exploited the properties of graphs of bounded treewidth and obtained a constant factor approximation algorithm, a $2^{\mathcal{O}(k \log k)} \cdot n$ time parameterized algorithm, and—most importantly, from our perspective—a polynomial sized kernel for every Planar $\mathcal{F}$-Minor-Free Deletion problem. More precisely, they showed that Planar $\mathcal{F}$-Minor-Free Deletion admits a kernel of size $f(\mathcal{F}) \cdot k^{g(\mathcal{F})}$ for some functions f and g. The degree g of the polynomial in the kernel size grows very quickly; it is not even known to be computable. This result is the starting point of our research.

> Does Planar $\mathcal{F}$-Minor-Free Deletion have kernels whose size is *uniformly polynomial*, of the form $f(\mathcal{F}) \cdot k^c$ for a universal constant c that does not depend on $\mathcal{F}$?

We prove that some families of Planar $\mathcal{F}$-Minor-Free Deletion problems *do not* have uniformly polynomial kernels (unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$). Since a graph class has bounded treewidth if and only if it excludes a planar graph as a minor, a canonical Planar $\mathcal{F}$-Minor-Free Deletion problem is Treewidth-η Deletion: can k vertices be removed to obtain a graph of treewidth at most η? We denote by K_d and P_d a clique and path on d vertices, respectively. Our first theorem is the following lower bound result.

Theorem 1. *Let $d \geq 3$ be a fixed integer and $\epsilon > 0$. If the parameterization by solution size k of one of the problems*

1. $\{K_{d+1}\}$-Minor-Free Deletion,
2. $\{K_{d+1}, P_{4d}\}$-Minor-Free Deletion, *and*
3. Treewidth-$(d-1)$ Deletion

admits a compression of bitsize $\mathcal{O}(k^{\frac{d}{2}-\epsilon})$, or a kernel with $\mathcal{O}(k^{\frac{d}{4}-\epsilon})$ vertices, then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. In fact, even if the parameterization by the size x of a vertex cover of the input graph admits a compression of bitsize $\mathcal{O}(x^{\frac{d}{2}-\epsilon})$ or a kernel with $\mathcal{O}(x^{\frac{d}{4}-\epsilon})$ vertices, then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

Theorem 1 shows that the kernelization result of Fomin et al. [11] is tight in the following sense: the degree g of the polynomial in the kernel sizes for Planar $\mathcal{F}$-Minor-Free Deletion must depend on the family $\mathcal{F}$. In fact, the theorem gives the stronger result that even parameterized by the *vertex cover number* of the graph (a larger parameter), the Treewidth-η Deletion problem does not admit uniformly polynomial kernels unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. This resolves an open problem of Cygan et al. [5].

A graph class has bounded treewidth if and only if it excludes a planar graph as a minor. Thus, by restricting the $\mathcal{F}$-Minor-Free Deletion problem to those $\mathcal{F}$ that contain a planar graph, one exploits the properties of graphs of bounded treewidth to design polynomial kernels for Planar $\mathcal{F}$-Minor-Free Deletion. It is a natural question whether further restrictions on $\mathcal{F}$ lead to uniformly polynomial kernels. However, the second item of Theorem 1 shows that even when $\mathcal{F}$ contains a *path*, the degree of the polynomial must, in general, depend on the set $\mathcal{F}$. This raises the question whether there are any general families of $\mathcal{F}$-Minor-Free Deletion problems that admit uniformly polynomial kernels.

Excluding planar minors results in graphs of bounded treewidth [19]; excluding forest minors results in graphs of bounded pathwidth [18]; and excluding path minors results in graphs of bounded treedepth [16]. A canonical $\mathcal{F}$-Minor-Free Deletion problem when $\mathcal{F}$ contains a path is therefore:

Treedepth-η Deletion	**Parameter:** k
Input: An undirected graph G and a positive integer k.	
Question: Does there exist a subset $Z \subseteq V(G)$ of size at most k such that $\mathbf{td}(G - Z) \leq \eta$?	

Here $\mathbf{td}(G)$ denotes the treedepth of a graph G. The set Z is called a *treedepth-η modulator* of G. Surprisingly, we show that Treedepth-η Deletion admits uniformly polynomial kernels. More precisely, we obtain the following theorem.

Theorem 2. Treedepth-η Deletion *admits a kernel with* $2^{\mathcal{O}(\eta^2)}k^6$ *vertices.*

We prove several new results about the structure of optimal treedepth decompositions and exploit this to obtain the desired kernel for Treedepth-η Deletion. Unlike the kernelization algorithm of Fomin et al. [11], our kernel is completely explicit. It does not use the machinery of protrusion replacement, which was introduced to the context of kernelization by Bodlaender et al. [2] and has subsequently been applied in various scenarios [8,10,12,15]. Using protrusion replacement one can prove that kernelization algorithms exist, but the technique generally does not explicitly give the algorithm nor a concrete size bound for the resulting kernel.

Techniques. The kernelization lower bound of Theorem 1 is obtained by reduction from Exact d-Uniform Set Cover, parameterized by the number of sets in the solution. Existing lower bounds exist for these problems due to Dell and Marx [6] and Hermelin and Wu [14], showing that the degree of the kernel size must grow linearly with the cardinality d of the sets in the input. While the construction that proves Theorem 1 is relatively simple in hindsight, the fact that the construction applies to all three mentioned problems, and also applies to the parameterization by vertex cover number, makes it interesting.

Our main technical contribution lies in the kernelization algorithm for Treedepth-η Deletion. Our algorithm starts by enriching the graph G by adding edges between vertices that are connected by many internally vertex-disjoint paths. Like in prior work on Treewidth-η Deletion [5], adding such edges does not change the answer to the problem. We then apply an algorithm by Reidl et al. [17] to compute an approximate treedepth-η modulator S of the resulting graph. The remainder of the algorithm strongly exploits the structure of the bounded-treedepth graph $G - S$. By combining separators for vertices that are not linked through many disjoint paths, we compute a small set Y such that all the bounded-treedepth connected components of $G - (S \cup Y)$ have a special structure: their neighborhood in S forms a clique, while they have less than η neighbors in Y. For such components C we can prove that optimal treedepth-η modulators contain at most 2η vertices from C. This important fact allows us to infer that optimal solutions cannot disturb the structure of the graph $G[C]$ too much. While it is relatively easy to bound the number of connected components of $G - (S \cup Y)$, the main work consists of reducing the size of each such component.

We formulate three lemmata that analyze under which circumstances the structure of optimal treedepth-η modulators is preserved when adding edges, removing edges, and removing vertices of the graph. By exploiting the fact that the solution size within a particular part C of the graph is constant, these lemmata ensure that even after deleting an optimal modulator from C, the remainder of C forces a structure of treedepth decompositions of the remaining graph that is compatible with the graph modifications. Of particular interest is the lemma showing that if v dominates the neighborhood of component C, then edges of v into the component may be safely discarded if certain other technical conditions are met.

The three described lemmata are the main tool in the reduction algorithm. To shrink components of $G-(S\cup Y)$ we have to add some edges, while removing other edges, to create settings where vertices can be removed from the instance without changing its answer. The fact that we have to combine edge additions and removals makes our reduction algorithm quite delicate: we cannot simply formulate reduction rules for adding and removing edges and apply them exhaustively, as they would work against each other. We therefore present a recursive algorithm that processes a treedepth-η decomposition of $G-S$ from top to bottom, making suitable transformations that bound the degree of the modulator S into the remainder of the component C. Using a careful measure expressed in terms of this degree, we can then prove that our algorithm achieves the desired size reduction.

Related Results. Planar $\mathcal{F}$-Minor-Free Deletion has received considerable attention [11,15] resulting in approximation, kernelization, and FPT algorithms. Cygan et al. [5] studied Treewidth-η Deletion parameterized by the vertex cover number of a graph and obtained a kernel of size $k^{\mathcal{O}(\eta)}$. In a later paper, Fomin et al. [9] studied $\mathcal{F}$-Minor-Free Deletion parameterized by the vertex cover number of the graph. They obtained kernels of size $k^{\mathcal{O}(\Delta(\mathcal{F}))}$, where $\Delta(\mathcal{F})$ is an upper bound on the maximum degree of any graph in $\mathcal{F}$. Notable work involving the parameter treedepth includes the $2^{\mathcal{O}(t^2)}\cdot n$-time algorithm for testing treedepth by Reidl et al. [17] and the kernelization meta-theorems for problems parameterized by a treedepth-η modulator by Gajarský et al. [12].

2 Preliminaries

Notation not defined here is standard. All graphs we consider are finite, undirected, and simple. We write $H \subseteq G$ if H is a subgraph of G. Given two distinct vertices u and v we define $\lambda_G(u,v)$ as the maximum cardinality of a set of pairwise internally vertex-disjoint uv-paths in the graph G.

Treedepth. A *rooted tree* T is a tree with one distinguished vertex $r \in V(T)$, called the *root* of T. A *rooted forest* is a disjoint union of rooted trees. The roots introduce natural parent-child and ancestor-descendant relations between vertices in forest. A vertex x is a proper ancestor (proper descendant) of a vertex y if x is an ancestor (descendant) of y and $x \neq y$. We denote by $\mathbf{anc}_F(x)$

the proper ancestors of x; this set is empty if x is a root. We denote by $\pi(x)$ the parent of x in F. The parent of the root of the tree is $\perp$. For a rooted forest F and a vertex $v \in V(F)$, we denote by F_v the subtree rooted at v that contains all v's descendants, including v itself. The *depth* of a vertex x in a rooted forest F is the number of vertices on the unique simple path from x to the root of the tree to which x belongs; it is denoted $\mathbf{depth}(x, F)$. The *height* of v is the maximum number of vertices on a simple path from v to a leaf in F_v. The height of F is the maximum height of a vertex of F and is denoted $\mathbf{height}(F)$. Two vertices x and y are in *ancestor-descendant* relation if x is an ancestor of y or vice versa.

Definition 1 (Treedepth) *A treedepth decomposition of a graph G is a rooted forest F on the vertex set $V(G)$ (i.e., $V(G) = V(F)$) such that for every edge $\{u, v\}$ of G, the endpoints u and v are in ancestor-descendant relation. The* treedepth *of G, denoted $\mathbf{td}(G)$, is the least $d \in \mathbb{N}$ such that there exists a treedepth decomposition F of G with $\mathbf{height}(F) = d$.*

The following properties follow from this definition. The treedepth of a disconnected graph is the maximum treedepth of its connected components. If F is a treedepth decomposition of G and $S \subseteq V(G)$ induces a clique in G, then there is one root-to-leaf path in F containing all vertices of S. If H is a connected subgraph of G, then all vertices of H belong to the same tree in any treedepth decomposition. If $u, v \in V(H)$ are not in ancestor-descendant relation in T, then some vertex of H is a common ancestor of u and v.

We will work with the notion of a *nice treedepth decomposition.* A treedepth decomposition F of a graph G is a nice treedepth decomposition if, for every $v \in V(F)$, the subgraph of G induced by the vertices in F_v is connected. The following lemma shows that any graph has a minimum-height treedepth decomposition that is also nice.

Lemma 1 ([17]). *For every fixed η there is a polynomial-time algorithm that, given a graph G, either determines that $\mathbf{td}(G) > \eta$ or computes a nice treedepth decomposition F of G of depth $\mathbf{td}(G)$.*

Lemma 2 ([12, Lemma 2]). *Fix $\eta \in \mathbb{N}$. Given a graph G, one can in polynomial time compute a subset $S \subseteq V(G)$ such that $\mathbf{td}(G - S) \leq \eta$ and $|S|$ is at most 2^η times the size of a minimum treedepth-η modulator of G.*

3 Kernelization Lower Bounds

We turn our attention to kernelization and compression lower bounds. To prove that $\mathcal{F}$-Minor-Free Deletion does not have uniformly polynomial kernels for suitable families $\mathcal{F}$, we give a polynomial-parameter transformation from a problem for which a compression lower bound is known. The following problem is the starting point for our transformation.

Exact d-Uniform Set Cover **Parameter:** The universe size n. **Input:** A finite set U of size n, an integer k, and a set family $\mathcal{F} \subseteq 2^U$ of size-d subsets of U. **Question:** Is there a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ consisting of at most k sets such that every element of U is contained in exactly one subset of $\mathcal{F}'$?	

Observe that since all subsets in $\mathcal{F}$ have size exactly d, the requirement that each universe element is contained in exactly one subset in $\mathcal{F}'$ implies that a set $\mathcal{F}'$ can only be a solution if it consists of n/d subsets. This implies that $k = n/d$ for all nontrivial instances of the problem. Hermelin and Wu [14] obtained a compression lower bound for Exact d-Uniform Set Cover. The same problem was also studied by Dell and Marx [6] under the name Perfect d-Set Matching. They obtained a slightly stronger compression lower bound, which forms the starting point for our reduction.

Theorem 3 ([6, Theorem 1.2]). *For every fixed $d \geq 3$ and $\epsilon > 0$, there is no compression of size $\mathcal{O}(k^{d-\epsilon})$ for* Exact d-Uniform Set Cover *unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

We remark that, while Dell and Marx stated their main theorem in terms of kernelizations, the same lower bounds indeed hold for compressions. We present the construction that will be used to prove Theorem 1.

Lemma 3. *For every fixed d there is a polynomial-time algorithm that, given a set U of size n, an integer k, and a d-uniform set family $\mathcal{F} \subseteq \binom{U}{d}$, computes a graph G' with vertex cover number $\mathcal{O}(k^2)$ and an integer $k' \in \mathcal{O}(k^2)$, such that:*

1. *If there is a set $S' \subseteq V(G')$ of size at most k' such that $G' - S'$ is K_{d+1}-minor-free, then there is an exact set cover of U consisting of k sets from $\mathcal{F}$.*
2. *If there is an exact set cover of U consisting of k sets from $\mathcal{F}$, then there is a set $S' \subseteq V(G')$ of size at most k' such that $G' - S'$ is K_{d+1}-minor-free, P_{4d}-minor-free, and has treewidth at most $d - 1$.*

Proof. Given U of size n, the integer k, and the d-uniform set family $\mathcal{F}$, the algorithm proceeds as follows. If $k \neq n/d$ then no exact set cover with k sets exists; we output $G' := K_{d+1}$ and $k' := 0$. We focus on the case that $k = n/d$. The main idea behind the construction is to create an $n \times k$ matrix with one vertex per cell. Each one of the k columns contains n vertices that correspond to the n universe elements. By turning columns into cliques and adding small gadgets, we will ensure that solutions to the vertex deletion problem must take the following form: they delete all vertices of the matrix except for exactly d per column. By enforcing that from each row, all vertices but one are deleted, and that the d surviving vertices in a column form a subset in $\mathcal{F}$, we relate the minor-free deletion sets to solutions of the exact covering problem. The formal construction proceeds as follows. Without loss of generality we can assume that the universe U consists of $[n] = \{1, 2, \ldots, n\}$, which simplifies the exposition.

1. Initialize G' as the graph consisting of $n \times k$ vertices $v_{i,j}$ for $i \in [n]$ and $j \in [k]$. For each column index $j \in [k]$ turn the vertex set $\{v_{i,j} \mid i \in [n]\}$ into a clique. We refer to $M := \{v_{i,j} \mid i \in [n], j \in [k]\}$ as the *matrix vertices*.
2. For every row index $i \in [n]$ add a dummy clique D_i consisting of $d-1$ vertices to G'. Make all vertices in D_i adjacent to vertices $\{v_{i,j} \mid j \in [k]\}$ of the i-th row.
3. As the last step we encode the set family $\mathcal{F}$ into the graph. For every set $X \in \binom{U}{d} \setminus \mathcal{F}$, which is a size-$d$ subset of $[n]$ that is not in the set family $\mathcal{F}$, we do the following. For each column index $j \in [k]$, we create an *enforcer vertex* $f_{j,X}$ for the set X into column j. The neighborhood of $f_{j,X}$ consists of the d vertices $\{v_{i,j} \mid i \in X\}$, i.e., the vertices in column j corresponding to set X.

Observation 3.1. *$M \cup (\bigcup_{i \in n} D_i)$ is a vertex cover of G' of size $n(k+d) \in \mathcal{O}(k^2)$.*

This concludes the construction of G'. It is easy to see that it can be performed in polynomial time for fixed d, since G' has $\mathcal{O}(n^{d+1})$ vertices. Define $k' := k(n-d)$. Since d is fixed we may absorb it into the $\mathcal{O}$-notation. As $n = kd$ this implies $k' \in \mathcal{O}(k^2)$. The proof that the construction satisfies the desired properties is deferred to the full version. □

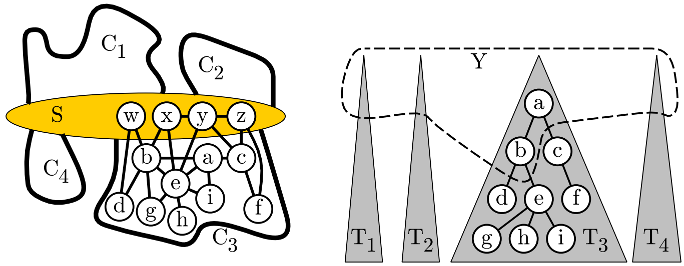
The proof of Theorem 1 follows by combining Lemma 3 with standard kernelization lower bound tools and Theorem 3. It can be found in the full version [13].

4 Uniformly Polynomial Kernelization for Treedepth-η Deletion

In this section we discuss the kernelization procedure for Treedepth-η Deletion. As this material spans twenty pages, space limitations prohibit us from giving full details here. For this extended abstract we have therefore chosen to give an intuitive high-level overview of the exploited structure and the preprocessing algorithm; details can be found in the full version [13]. As described in the introduction, the two main ingredients are a decomposition algorithm and a reduction algorithm, to be applied to each piece of the decomposition. Throughout this section, the reader should be aware of the two uses for the word decomposition employed here: on the one hand we are decomposing the input instance (G, k) of the deletion problem into several subgraphs that have a certain structure, while on the other hand the deletion problem we are solving asks for a set $S \subseteq V(G)$ whose removal ensures that $G - S$ has a bounded-height treedepth decomposition.

4.1 Structural Decomposition of the Input Graph

The first step of the decomposition phase enriches the input instance (G, k) with extra edges. In an analogue of previous work on Treewidth-η Deletion [5], we show that when there are non-adjacent vertices u and v in G such that $\lambda_G(u, v) \geq$



(a) Graph G, modulator S. (b) Decomposition of $G - S$.

Fig. 1. Schematic illustration of an instance that has been decomposed. 1(a) The resulting graph G and the suboptimal treedepth-4 modulator S in G used when decomposing. Graph $G - S$ has four connected components, of which the third is drawn in detail. 1(b) Illustration of the treedepth-4 decomposition F of $G - S$. The forest F contains four decomposition trees $T_1, \ldots, T_4$, one for each component of $G-S$. By the properties of a treedepth decomposition, for any vertex $v \in V(G) \setminus S$, each neighbor $u \in N_G(v)$ is an ancestor of v in F, descendant of v in F, or contained in S. The decomposition ensures that for each connected component C of $G - (S \cup Y)$, the set $N_G(C) \cap S$ is a clique. This is illustrated for the connected component consisting of $\{e, g, h, i\}$, whose neighbors among S are $\{x, y\}$, a 2-clique. As the set Y is closed under taking ancestors, it consists of the top parts of decomposition trees in F.

$k + \eta$, then adding the edge $\{u, v\}$ does not change the answer to the instance. After exhaustively adding connecting pairs for which this condition is satisfied, we are guaranteed that for any remaining non-adjacent pair of vertices $\{u, v\}$ in G we have $\lambda_G(u, v) < k + \eta$. By Menger's theorem, this implies that there is a uv-vertex separator of size less than $k + \eta$.

After enriching the graph, we use the polynomial-time approximation algorithm for Treedepth-η Deletion of Lemma 2 to find a suboptimal treedepth-η modulator S in the input instance (G, k) of size $\mathcal{O}(k)$. We use the structure that this modulator reveals in the bounded-treedepth subgraph $G - S$ to guide further processing, and compute a treedepth-η decomposition F of $G - S$ using Lemma 1. For every pair $\{u, v\}$ of remaining non-adjacent vertices in S, we compute a minimum uv-separator Y_{uv} and add $Y_{uv} \setminus S$ to a set Y. Since there are $\mathcal{O}(k^2)$ pairs of vertices among S, by the earlier bound this yields a set Y of size $\mathcal{O}(k^2(k + \eta))$. We then add all F-ancestors of vertices in Y to the set Y. Since each vertex has less than η ancestors in a treedepth-η decomposition, the size of Y increases by at most a factor η and remains $\mathcal{O}(k^3)$.

The resulting sets S and Y decompose the graph in a useful way. For every connected component C of $G - (S \cup Y)$, we know that $N_G(C) \cap S$ is a clique, since Y contains separators for all pairs of non-adjacent vertices in S. In addition, for every such component C we have $|N_G(C) \cap Y| < \eta$ since all such neighbors are contained on one root-to-leaf path of the height-η decomposition F. All such components C are therefore what we call *η-nearly clique separated*: there is a

Algorithm 1 Reduce(Graph G, treedepth-η modulator S, treedepth-η decomposition F of $G - S$, node v of F, $k \in \mathbb{N}$)

Let T be the tree in F containing v
while $\exists p, q \in N_G(T_v) \cup \{v\}$ with $\{p, q\} \notin E(G)$ and $\lambda_{G[\{p,q\} \cup T_v]}(p, q) \geq 3\eta$ **do**
    Add the edge $\{p, q\}$ to G
while $\exists$ distinct children $c_0, c_1, \ldots, c_{3\eta}$ of v s.t. c_0 has a neighbor $s \in S$, $N_G(T_{c_0}) \subseteq N_G[s]$, and for $i \in [3\eta]$ we have $\mathbf{td}(G[T_{c_i}]) \geq \mathbf{td}(G[T_{c_0}])$ and $s \in N_G(T_{c_i})$ **do**
    Remove the edges between s and members of T_{c_0} from graph G
while $\exists$ a child c^* of v such that $N_G(T_{c^*})$ is a clique, and for every $w \in N_G(T_{c^*})$ there are 3η distinct children $c_1^w, \ldots, c_{3\eta}^w \neq c^*$ of v such that for all $i \in [3\eta]$ we have $\mathbf{td}(G[T_{c_i^w}]) \geq \mathbf{td}(G[T_{c^*}])$ and $w \in N_G(T_{c_i^w})$ **do**
    Remove the vertices in T_{c^*} from F and from G
for each remaining child c of v in T **do**
    Reduce(G, S, F, c, k)

clique in G containing all but η vertices of $N_G(C)$. We prove that minimum treedepth-η modulators contain at most 2η vertices of such components.[1]

The fact that minimum solutions delete at most 2η vertices (a constant independent of k) from components C of $G - (S \cup Y)$ will be extremely useful later on. The last part of the decomposition phase bounds the number of connected components of $G - (S \cup Y)$. The number of *non-simplicial components* (components whose S-neighborhood is not a clique) is already $\mathcal{O}(k^2(k + \eta))$, since each component provides a path between non-adjacent vertices $\{u, v\} \in \binom{S}{2}$ for which $\lambda_G(u, v) < \eta + k$. To bound the *simplicial components* (those with $N_G(C) \cap S$ a clique) requires more work. We give a structural lemma showing how to find a simplicial component whose deletion does not change the answer to the problem, in the case that there are many of such components. This step is inspired by earlier work [3, Rule 6] on PATHWIDTH. The resulting reduced graph is given as the output of the decomposition phase, together with the suboptimal modulator S and the treedepth-η decomposition F of $G - S$. See Fig. 1 for a schematic illustration.

4.2 Reduction Algorithm

After the decomposition phase, the goal of the reduction phase is to shrink the size of the connected components of $G - (S \cup Y)$; since S and Y have size $\mathcal{O}(k)$ and $\mathcal{O}(k^3)$, respectively, and the number of components of $G - (S \cup Y)$ is also bounded uniformly polynomially in k, bounding the size of each such component suffices to bound the size of G. Using the notion of a *nice* treedepth decomposition, we can ensure that the connected components of $G - (S \cup Y)$ correspond to

[1] If a solution S contains more than 2η vertices from C, then one would get a smaller solution by leaving C untouched and instead deleting the at most η vertices of $N_G(C)$ that are not part of the clique, and the vertices of the clique in $N_G(C)$ that are not deleted by S; there are at most η of the latter since treedepth-η graphs contain no $\eta + 1$-cliques.

the vertex sets of subtrees of the decomposition forest F rooted at vertices that are not in Y, but whose parent is in Y. Observe that if we could ensure that the maximum degree in the decomposition forest F is bounded by some function of η (but independent of k), then we would immediately get a size bound as desired: any subtree of maximum degree $f(\eta)$ has at most $f(\eta)^\eta$ vertices, since its height is at most η. Such a degree reduction is therefore our goal. However, we are not able to bound the degree by a function that is independent of k. Instead, by a top-down reduction algorithm on the decomposition forest F we can guarantee that the degree of a node v in the decomposition forest F, is bounded linearly in $|N_G(F_v) \cap S|$, which is the number of vertices of S that are adjacent to a node in the subtree of F rooted at v. This fact alone is not sufficient to bound the sizes of components C of $G-(S \cup Y)$ by a polynomial of degree independent of η, for it does not rule out the possibility of a complete degree-$|S|$ tree of height η, containing $\Omega(k^\eta)$ nodes.

The main challenge in obtaining uniformly polynomial kernels is to overcome this obstacle. To do so, we go through the decomposition trees from top to bottom, at every stage reducing the degree of the current node v using three new structural insights on treedepth. We ensure that every vertex $s \in S$ that has neighbors in any subtree rooted at a child of v, has a neighbor in at most $2^\eta \cdot 3\eta$ subtrees rooted at children of v. If this is violated, then we can first introduce new edges from s to ancestors of v and other members of S using an edge addition lemma, and afterward discard edges from s to descendants of v using an edge deletion lemma. Then we reduce the number of children whose subtrees contain no neighbors of S to constant, using a vertex deletion lemma. The procedure achieving this is given as the Reduce algorithm; its initial call is for the node v for which F_v contains the nodes of the component C we are shrinking. A careful induction reveals that this process is successful in reducing the total number of nodes in a connected component C of $G-(S \cup Y)$ to $f(\eta) \cdot k$. This achieves the desired total size reduction and yields a proof of Theorem 2.

5 Conclusion

In this paper we (re-)studied the Planar $\mathcal{F}$-Minor-Free Deletion problem from the perspective of (uniform) kernelization. We answered the question whether all Planar $\mathcal{F}$-Minor-Free Deletion problems have uniformly polynomial kernels negatively, but showed that the special case Treedepth-η Deletion (which is a Planar $\mathcal{F}$-Minor-Free Deletion problem for every η, where every $\mathcal{F}$ contains a path) has uniformly polynomial kernels.

The distinction between uniformly versus non-uniformly polynomial kernels is similar to the distinction between algorithms whose parameter dependence is fixed-parameter tractable (FPT) versus slicewise-polynomial (XP), and opens up a similarly broad area of investigation. The kernelization complexity of $\mathcal{F}$-Minor-Free Deletion is still wide open. Some notable open problems in this direction are: (1) Does $\mathcal{F}$-Minor-Free Deletion admit a polynomial kernel for any fixed set $\mathcal{F}$, even when $\mathcal{F}$ contains no planar graphs? Even for the special case of deleting k vertices to get a planar graph (Vertex Planarization), we do not know the answer. (2) Is it possible to obtain a dichotomy theorem, characterizing the families $\mathcal{F}$ for which Planar $\mathcal{F}$-Minor-Free Deletion admits uniformly polynomial kernels? These questions are part of a large research program into the complexity of $\mathcal{F}$-Minor-Free Deletion problems, whose importance was recognized by its listing in the *Research Horizons* section of the recent textbook by Downey and Fellows [7, Chapter 33.2].

References

1. Adler, I., Grohe, M., Kreutzer, S.: Computing excluded minors. In: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008), pp. 641–650. ACM-SIAM (2008)
2. Bodlaender, H., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) Kernelization. In: Proc. 50th FOCS, pp. 629–638. IEEE (2009)
3. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Kernel bounds for structural parameterizations of pathwidth. In: Fomin, F.V., Kaski, P. (eds.) SWAT 2012. LNCS, vol. 7357, pp. 352–363. Springer, Heidelberg (2012)
4. Chekuri, C., Chuzhoy, J.: Polynomial bounds for the grid-minor theorem. In: Proc. 46th STOC, pp. 60–69 (2014)
5. Cygan, M., Lokshtanov, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: On the hardness of losing width. In: Marx, D., Rossmanith, P. (eds.) IPEC 2011. LNCS, vol. 7112, pp. 159–168. Springer, Heidelberg (2012)
6. Dell, H., Marx, D.: Kernelization of packing problems. In: Proc. 23rd SODA, pp. 68–81 (2012)
7. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer (2013)
8. Fomin, F., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Bidimensionality and kernels. In: Proc. 21st SODA, pp. 503–510 (2010)
9. Fomin, F.V., Jansen, B.M.P., Pilipczuk, M.: Preprocessing subgraph and minor problems: When does a small vertex cover help? J. Comput. System Sci. **80**(2), 468–495 (2014)
10. Fomin, F.V., Lokshtanov, D., Misra, N., Philip, G., Saurabh, S.: Hitting forbidden minors: approximation and kernelization. In: Proc. 28th STACS, pp. 189–200 (2011)
11. Fomin, F.V., Lokshtanov, D., Misra, N., Saurabh, S.: Planar $\mathcal{F}$-Deletion: approximation, kernelization and optimal FPT algorithms. In: Proc. 53rd FOCS, pp. 470–479 (2012)
12. Gajarský, J., Hliněný, P., Obdržálek, J., Ordyniak, S., Reidl, F., Rossmanith, P., Sánchez Villaamil, F., Sikdar, S.: Kernelization using structural parameters on sparse graph classes. In: Bodlaender, H.L., Italiano, G.F. (eds.) ESA 2013. LNCS, vol. 8125, pp. 529–540. Springer, Heidelberg (2013)
13. Giannopoulou, A.C., Jansen, B.M.P., Lokshtanov, D., Saurabh, S.: Uniform kernelization complexity of hitting forbidden minors (2015). CoRR, abs/1502.03965
14. Hermelin, D., Wu, X.: Weak compositions and their applications to polynomial lower bounds for kernelization. In: Proc. 23rd SODA, pp. 104–113 (2012)
15. Kim, E.J., Langer, A., Paul, C., Reidl, F., Rossmanith, P., Sau, I., Sikdar, S.: Linear kernels and single-exponential algorithms via protrusion decompositions. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) ICALP 2013, Part I. LNCS, vol. 7965, pp. 613–624. Springer, Heidelberg (2013)

16. Nesetril, J., Ossona de Mendez, P.: Tree-depth, subgraph coloring and homomorphism bounds. Eur. J. Comb. **27**(6), 1022–1041 (2006)
17. Reidl, F., Rossmanith, P., Villaamil, F.S., Sikdar, S.: A faster parameterized algorithm for treedepth. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014. LNCS, vol. 8572, pp. 931–942. Springer, Heidelberg (2014)
18. Robertson, N., Seymour, P.D.: Graph minors. I. Excluding a forest. J. Comb. Theory, Ser. B **35**(1), 39–61 (1983)
19. Robertson, N., Seymour, P.D.: Graph minors. V. Excluding a planar graph. J. Combin. Theory Ser. B **41**(1), 92–114 (1986)
20. Robertson, N., Seymour, P.D.: Graph minors. XIII. The disjoint paths problem. J. Combin. Theory Ser. B **63**(1), 65–110 (1995)