Linear Kernels and Single-Exponential Algorithms via Protrusion Decompositions*

Eun Jung Kim[1], Alexander Langer[2], Christophe Paul[3], Felix Reidl[2], Peter Rossmanith[2], Ignasi Sau[3], and Somnath Sikdar[2]

[1] CNRS, LAMSADE, Paris, France
eunjungkim78@gmail.com
[2] Theoretical Computer Science, Department of Computer Science, RWTH Aachen University, Germany
{langer,reidl,rossmani,sikdar}@cs.rwth-aachen.de
[3] CNRS, LIRMM, Montpellier, France
{paul,sau}@lirmm.fr

Abstract. We present a linear-time algorithm to compute a decomposition scheme for graphs G that have a set $X \subseteq V(G)$, called a *treewidth-modulator*, such that the treewidth of $G - X$ is bounded by a constant. Our decomposition, called a *protrusion decomposition*, is the cornerstone in obtaining the following two main results. Our first result is that any parameterized graph problem (with parameter k) that has *finite integer index* and such that positive instances have a treewidth-modulator of size $O(k)$ admits a linear kernel on the class of H-topological-minor-free graphs, for any fixed graph H. This result partially extends previous meta-theorems on the existence of linear kernels on graphs of bounded genus and H-minor-free graphs.

Let $\mathcal{F}$ be a fixed finite family of graphs containing at least one planar graph. Given an n-vertex graph G and a non-negative integer k, Planar-$\mathcal{F}$-Deletion asks whether G has a set $X \subseteq V(G)$ such that $|X| \leqslant k$ and $G - X$ is H-minor-free for every $H \in \mathcal{F}$. As our second application, we present the first *single-exponential* algorithm to solve Planar-$\mathcal{F}$-Deletion. Namely, our algorithm runs in time $2^{O(k)} \cdot n^2$, which is asymptotically optimal with respect to k. So far, single-exponential algorithms were only known for special cases of the family $\mathcal{F}$.

Keywords: parameterized complexity, linear kernels, algorithmic meta-theorems, sparse graphs, single-exponential algorithms, graph minors.

1 Introduction

This work contributes to the two main areas of parameterized complexity, namely, kernels and fixed-parameter tractable (FPT) algorithms (see, e.g., [11] for an introduction). In many cases, the key ingredient in order to solve a hard graph

* This article replaces and extends the results of [CoRR, abs/1201.2780, 2012]. Research funded by DFG-Project RO 927/12-1 "Theoretical and Practical Aspects of Kernelization", ANR project AGAPE (ANR-09-BLAN-0159), and the Languedoc-Roussillon Project "Chercheur d'avenir" KERNEL.

F.V. Fomin et al. (Eds.): ICALP 2013, Part I, LNCS 7965, pp. 613–624, 2013.

problem is to find an appropriate *decomposition* of the input graph, which allows to take advantage of the structure given by the graph class and/or the problem under study. In this article we follow this paradigm and present (in Section 3) a novel linear-time algorithm to compute a decomposition for graphs G that have a set $X \subseteq V(G)$, called *t-treewidth-modulator*, such that the treewidth of $G - X$ is at most some constant $t - 1$. We then exploit this decomposition in two different ways: to *analyze* the size of kernels and to obtain *efficient* FPT algorithms. We would like to note that similar decompositions have already been (explicitly or implicitly) used for obtaining polynomial kernels [1, 4, 13, 15, 18].

Linear Kernels. During the last decade, a plethora of results emerged on linear kernels for graph-theoretic problems restricted to *sparse* graph classes. A celebrated result by Alber *et al.* [1] prompted an explosion of research papers on linear kernels on planar graphs. Guo and Niedermeier [18] designed a general framework and showed that problems that satisfy a certain "distance property" have linear kernels on planar graphs. Bodlaender *et al.* [4] provided a meta-theorem for problems to have a linear kernel on graphs of bounded genus. Fomin *et al.* [15] extended these results for bidimensional problems on H-minor-free graphs. A common feature of these meta-theorems on sparse graphs is a *decomposition scheme* of the input graph that, loosely speaking, allows to deal with each part of the decomposition independently. For instance, the approach of [18], which is much inspired from [1], is to consider a so-called *region decomposition* of the input planar graph. The key point is that in an appropriately reduced YES-instance, there are $O(k)$ regions and each one has constant size, yielding the desired linear kernel. This idea was generalized in [4] to graphs on surfaces, where the role of regions is played by *protrusions*, which are graphs with small treewidth and small boundary (see Section 2 for details). The resulting decomposition is called *protrusion decomposition*. A crucial point is that while the reduction rules of [1] are *problem-dependent*, those of [4] are *automated*, relying on a property called *finite integer index* (FII), which was introduced by Bodlaender and de Fluiter [5]. Having FII essentially guarantees that "large" protrusions of an instance can be replaced by "small" equivalent gadget graphs. This operation is usually called the *protrusion replacement rule*. FII is also of central importance to the approach of [15] on H-minor-free graphs.

In the spirit of the above results, our algorithm to compute protrusion decompositions allows us to prove that we can obtain (in Section 4) linear kernels on a larger class of sparse graphs. A parameterized problem is *treewidth-bounding* if YES-instances have a t-treewidth-modulator of size $O(k)$ for some constant t. Our first main result is:

Theorem I. Fix a graph H. Let Π be a parameterized graph problem on the class of H-topological-minor-free graphs that is treewidth-bounding and has FII. Then Π admits a linear kernel.

It turns out that a host of problems including TREEWIDTH-t VERTEX DELETION, CHORDAL VERTEX DELETION, INTERVAL VERTEX DELETION, EDGE DOMINATING SET, to name a few, satisfy the conditions of our theorem. Since for any fixed graph H, the class of H-topological-minor-free graphs strictly contains the class of H-minor-free graphs, our result is in fact an extension of the results of Fomin *et al.* [15].

Efficient FPT algorithms. In the second part of the paper (Section 5) we are interested in *single-exponential* algorithms, that is, algorithms that solve a parameterized problem with parameter k on an n-vertex graph in time $2^{O(k)} \cdot n^{O(1)}$. Let $\mathcal{F}$ be a finite family of graphs containing at least one planar graph. In the Planar-$\mathcal{F}$-Deletion problem, given a graph G and a non-negative integer parameter k as input, we are asked whether G has a set $X \subseteq V(G)$ such that $|X| \leqslant k$ and $G - X$ is H-minor-free for every $H \in \mathcal{F}$.

Note that Vertex Cover and Feedback Vertex Set correspond to the special cases of $\mathcal{F} = \{K_2\}$ and $\mathcal{F} = \{K_3\}$, respectively. Recent works have provided, using quite different techniques, single-exponential algorithms for the particular cases $\mathcal{F} = \{K_3, T_2\}$ [7,22], $\mathcal{F} = \{\theta_c\}$ [19], or $\mathcal{F} = \{K_4\}$ [20]. The Planar-$\mathcal{F}$-Deletion problem was first stated by Fellows and Langston [12], who proposed a non-uniform $f(k) \cdot n^2$-time algorithm for some function $f(k)$, relying on the meta-theorem of Robertson and Seymour [24]. Explicit bounds on the function $f(k)$ can be obtained via dynamic programming. Indeed, as the Yes-instances of Planar-$\mathcal{F}$-Deletion have treewidth $O(k)$, using standard dynamic programming techniques on graphs of bounded treewidth (see for instance [2]), it can be seen that Planar-$\mathcal{F}$-Deletion can be solved in time $2^{2^{O(k \log k)}} \cdot n^2$. Recently, Fomin *et al.* [14] provided a $2^{O(k)} \cdot n \log^2 n$-time algorithm for the Planar-Connected-$\mathcal{F}$-Deletion problem, which is the special case of Planar-$\mathcal{F}$-Deletion when *every* graph in the family $\mathcal{F}$ is *connected*. In this paper we get rid of the connectivity assumption:

Theorem II. Planar-$\mathcal{F}$-Deletion can be solved in time $2^{O(k)} \cdot n^2$.

This result unifies, generalizes, and simplifies a number of results given in [6,8, 14,17,19,20]. Besides the fact that removing the connectivity constraint is an important theoretical step towards the general case where $\mathcal{F}$ may not contain any planar graph, it turns out that many natural such families $\mathcal{F}$ do contain disconnected planar graphs [10]. An important feature of our approach, in comparison with previous work [14,19,20], is that our algorithm *does not use any reduction rule*. This is because if $\mathcal{F}$ may contain disconnected graphs, Planar-$\mathcal{F}$-Deletion has not FII for some choices of $\mathcal{F}$, and then the protrusion replacement rule cannot be applied. A more in-depth discussion can be found in the full version. Finally, it should also be noted that the function $2^{O(k)}$ in Theorem II is best possible assuming the Exponential Time Hypothesis (ETH), as Vertex Cover cannot be solved in time $2^{o(k)} \cdot poly(n)$ unless the ETH fails.

Further research. Concerning our kernelization algorithms, a natural question is whether similar results can be obtained for an even larger class of sparse graphs. As discussed in the full version, obtaining a kernel for Treewidth-t Vertex Deletion on graphs of bounded expansion is as hard as on general graphs, and according to Fomin *et al.* [14], this problem has a kernel of size $k^{O(t)}$ on general graphs, and no uniform polynomial kernel (a polynomial kernel whose degree does not depend on t) is known. This fact makes us suspect that our kernelization result may settle the limit of meta-theorems about the existence of linear, or even uniform polynomial, kernels on sparse graph classes. We would like to note that the degree of the polynomial of the running time of our kernelization algorithm depends linearly on the size of the excluded topological minor H. It seems that the recent *fast protrusion replacer* of Fomin *et al.* [14] could be applied to get rid of this dependency on H.

Concerning the PLANAR-$\mathcal{F}$-DELETION problem, no single-exponential algorithm is known when the family $\mathcal{F}$ does not contain any planar graph. Is it possible to find such a family, or can it be proved that, under some complexity assumption, a single-exponential algorithm is not possible? Very recently, a randomized (Monte Carlo) constant-factor approximation algorithm for PLANAR-$\mathcal{F}$-DELETION has been given by Fomin *et al.* [14]. Finding a deterministic constant-factor approximation remains open.

2 Preliminaries

We use standard graph-theoretic notation (see [9] and the full version for any undefined terminology). Given a graph G, we let $V(G)$ denote its vertex set and $E(G)$ its edge set. A *minor* of G is a graph obtained from a subgraph of G by contracting zero or more edges. A *topological minor* of G is a graph obtained from a subgraph of G by contracting zero or more edges, such that each contracted edge has at least one endpoint with degree at most two. A graph G is *H-(topological)-minor-free* if G does not contain H as a (topological) minor.

A *parameterized graph problem* Π is a set of tuples (G,k), where G is a graph and $k \in \mathbb{N}_0$. If $\mathcal{G}$ is a graph class, we define Π *restricted to* $\mathcal{G}$ as $\Pi_{\mathcal{G}} = \{(G,k) \mid (G,k) \in \Pi \text{ and } G \in \mathcal{G}\}$. A parameterized problem Π is *fixed-parameter tractable* (FPT for short) if there exists an algorithm that decides instances (x,k) in time $f(k) \cdot poly(|x|)$, where f is a function of k alone. A *kernelization algorithm*, or just *kernel*, for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}_0$ is an algorithm that given $(x,k) \in \Gamma^* \times \mathbb{N}_0$ outputs, in time polynomial in $|x| + k$, an instance $(x',k') \in \Gamma^* \times \mathbb{N}_0$ such that $(x,k) \in \Pi$ if and only if $(x',k') \in \Pi$, and $|x'|, k' \leqslant g(k)$, where g is some computable function. The function g is called the *size* of the kernel. If $g(k) = k^{O(1)}$ or $g(k) = O(k)$, we say that Π admits a *polynomial kernel* and a *linear kernel*, respectively.

Given a graph $G = (V,E)$, we denote a *tree-decomposition of* G by $(T, \{W_x \mid x \in V(T)\})$, where T is a tree and $\{W_x \mid x \in V(T)\}$ are the bags of the decomposition. We refer the reader to Diestel's book [9] for an introduction to the theory of treewidth.

We restate the main definitions of the protrusion machinery developed in [4, 15]. Given a graph $G = (V,E)$ and a set $W \subseteq V$, we define $\partial_G(W)$ as the set of vertices in W that have a neighbor in $V \setminus W$. For a set $W \subseteq V$ the neighborhood of W is $N^G(W) = \partial_G(V \setminus W)$. Superscripts and subscripts are omitted when it is clear which graph is being referred to.

Given a graph G, a set $W \subseteq V(G)$ is a *t-protrusion* of G if $|\partial_G(W)| \leqslant t$ and $\mathbf{tw}(G[W]) \leqslant t-1$.[1] If W is a t-protrusion, the vertex set $W' = W \setminus \partial_G(W)$ is the *restricted protrusion of* W. We call $\partial_G(W)$ the *boundary* and $|W|$ the *size* of the t-protrusion W of G. Given a restricted t-protrusion W', we denote its *extended protrusion* by $W'^{+} = W' \cup N(W')$.

A *t-boundaried graph* is a graph $G = (V, E)$ with a set $\mathbf{bd}(G)$ (called the *boundary*[2] or the *terminals* of G) of t distinguished vertices labeled 1 through t. Let $\mathcal{G}_t$ denote the class of t-boundaried graphs, with graphs from $\mathcal{G}$. If $W \subseteq V$ is an r-protrusion in G, then we let G_W be the r-boundaried graph $G[W]$ with boundary $\partial_G(W)$, where the vertices of $\partial_G(W)$ are assigned labels 1 through r according to their order in G. *Gluing* two t-boundaried graphs G_1 and G_2 creates the graph $G_1 \oplus G_2$ obtained by taking the disjoint union of G_1 and G_2 and identifying each vertex in $\mathbf{bd}(G_1)$ with its corresponding vertex in $\mathbf{bd}(G_2)$, i.e. those vertices sharing the same label.

If G_1 is a subgraph of G with a t-boundary $\mathbf{bd}(G_1)$, *ungluing* G_1 *from* G creates the t-boundaried graph $G \ominus G_1 = G - (V(G_1) \setminus \mathbf{bd}(G_1))$ with boundary $\mathbf{bd}(G \ominus G_1) = \mathbf{bd}(G_1)$, the vertices of which are assigned labels according to their order in the graph G. Let W be a t-protrusion in G, let G_W denote the graph $G[W]$ with boundary $\mathbf{bd}(G_W) = \partial_G(W)$, and let G_1 be a t-boundaried graph. Then *replacing* G_W *by* G_1 corresponds to the operation $(G \ominus G_W) \oplus G_1$.

An (α, t)*-protrusion decomposition* of a graph G is a partition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ of $V(G)$ such that: (1) for every $1 \leqslant i \leqslant \ell$, $N(Y_i) \subseteq Y_0$; (2) $\max\{\ell, |Y_0|\} \leqslant \alpha$; (3) for every $1 \leqslant i \leqslant \ell$, $Y_i \cup N_{Y_0}(Y_i)$ is a t-protrusion of G. Y_0 is called the *separating part* of $\mathcal{P}$. Hereafter, the value of t will be fixed to some constant. When G is the input of a parameterized graph problem with parameter k, we say that an (α, t)-protrusion decomposition of G is *linear* whenever $\alpha = O(k)$.

Let $\Pi_{\mathcal{G}}$ be a parameterized graph problem restricted to a class $\mathcal{G}$ and let G_1, G_2 be two t-boundaried graphs in $\mathcal{G}_t$. We say that $G_1 \equiv_{\Pi,t} G_2$ if there exists a constant $\Delta_{\Pi,t}(G_1, G_2)$ (that depends on Π, t, and the ordered pair (G_1, G_2)) such that for all t-boundaried graphs G_3 and for all k: (1) $G_1 \oplus G_3 \in \mathcal{G}$ iff $G_2 \oplus G_3 \in \mathcal{G}$; (2) $(G_1 \oplus G_3, k) \in \Pi$ iff $(G_2 \oplus G_3, k + \Delta_{\Pi,t}(G_1, G_2)) \in \Pi$. We say that the problem $\Pi_{\mathcal{G}}$ has *finite integer index in the class* $\mathcal{G}$ iff for every integer t, the equivalence relation $\equiv_{\Pi,t}$ has finite index. In the case that $(G_1 \oplus G, k) \notin \Pi$ or $G_1 \oplus G \notin \mathcal{G}$ for *all* $G \in \mathcal{G}_t$, we set $\Delta_{\Pi,t}(G_1, G_2) = 0$. Note that $\Delta_{\Pi,t}(G_1, G_2) = -\Delta_{\Pi,t}(G_2, G_1)$.

If a parameterized problem has FII then it can be reduced by "replacing protrusions", hinging on the fact that each "large" protrusion can be replaced by a "small" gadget from the same equivalence class that behaves similar w.r.t. to the problem at hand. Exchanging G_1 by a gadget G_2 changes the parameter k by $\Delta_{\Pi,t}(G_1, G_2)$. Lemma 1 guarantees the existence of a set of representatives such that the replacement operation does *not* increase the parameter. In the full version we show how to find protrusions in polynomial time and how to identify by which representative to replace a protrusion, assuming that we are *given* the

[1] In [4], $\mathbf{tw}(G[W]) \leqslant t$, but we want the size of the bags to be at most t.

[2] Usually denoted by $\partial(G)$, but this collides with our usage of ∂.

set of representatives, an assumption we make from now on. This makes our algorithms in Section 4 *non-uniform*, as those in previous works [4, 13–15].

Lemma 1. [$\star$] [3] *Let Π be a parameterized graph problem that has FII in a graph class $\mathcal{G}$. Then for every t, there exists a finite set $\mathcal{R}_t$ of t-boundaried graphs such that for each $G \in \mathcal{G}_t$ there exists $G' \in \mathcal{R}_t$ such that $G \equiv_{\Pi,t} G'$ and $\Delta_{\Pi,t}(G, G') \geqslant 0$.*

For a parameterized problem Π that has FII in the class $\mathcal{G}$, let $\mathcal{R}_t$ denote the set of representatives as in Lemma 1. The *protrusion limit* of $\Pi_\mathcal{G}$ is a function $\rho_{\Pi_\mathcal{G}} : \mathbb{N} \to \mathbb{N}$ defined as $\rho_{\Pi_\mathcal{G}}(t) = \max_{G \in \mathcal{R}_t} |V(G)|$. We drop the subscript when it is clear which graph problem is being referred to. We also define $\rho'(t) := \rho(2t)$.

Lemma 2 ([4]). [$\star$] *Let Π be a parameterized graph problem with FII in $\mathcal{G}$ and let $t \in \mathbb{N}$ be a constant. For a graph $G \in \mathcal{G}$, if one is given a t-protrusion $X \subseteq V(G)$ such that $\rho'_{\Pi_\mathcal{G}}(t) < |X|$, then one can, in time $O(|X|)$, find a $2t$-protrusion W such that $\rho'_{\Pi_\mathcal{G}}(t) < |W| \leqslant 2 \cdot \rho'_{\Pi_\mathcal{G}}(t)$.*

3 Constructing Protrusion Decompositions

We present our algorithm to compute protrusion decompositions. Algorithm 1 marks the bags of a tree-decomposition of an input graph G that comes equipped with a t-treewidth-modulator $X \subseteq V(G)$. Our algorithm also takes an additional integer parameter r, which depends on the graph class to which G belongs and the precise problem one might want to solve (see Sections 4 and 5 for details).

Note that an optimal tree-decomposition of every connected component C of $G - X$ such that $|N_X(C)| \geqslant r$ can be computed in time linear in $n = |V(G)|$ using the algorithm of Bodlaender [3]. In the full version we sketch how the Large-subgraph marking step can be implemented using standard dynamic programming techniques. It is quite easy to see that Algorithm 1 runs in linear time.

Lemma 3. [$\star$] *Let Y_0 be the set of vertices computed by Algorithm 1. Every connected component C of $G - Y_0$ satisfies $|N_X(C)| < r$ and $|N_{Y_0}(C)| < r + 2t$, and thus forms a restricted protrusion.*

Given a graph G and a subset $S \subseteq V(G)$, we define a *cluster of $G - S$* as a maximal collection of connected components of $G - S$ with the same neighborhood in S. Note that the set of all clusters of $G - S$ induces a partition of the set of connected components of $G - S$, which can be easily found in linear time if G and S are given. By Lemma 3 and using the fact that $\mathbf{tw}(G - X) \leqslant t - 1$, the following proposition follows.

Proposition 1. *Let r, t be two positive integers, let G be a graph and $X \subseteq V(G)$ such that $\mathbf{tw}(G - X) \leqslant t - 1$, let $Y_0 \subseteq V(G)$ be the output of Algorithm 1 with input (G, X, r), and let $Y_1, \ldots, Y_\ell$ be the set of all clusters of $G - Y_0$. Then $\mathcal{P} := Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ is a $(\max\{\ell, |Y_0|\}, 2t + r)$-protrusion decomposition of G.*

[3] The proofs of the results marked with '[$\star$]' can be found in [CoRR, abs/1207.0835].

Algorithm 1: BAG MARKING ALGORITHM

Input: A graph G, a subset $X \subseteq V(G)$ such that $\mathbf{tw}(G-X) \leqslant t-1$, and an integer $r > 0$.

Set $\mathcal{M} \leftarrow \emptyset$ as the set of marked bags;
Compute an optimal rooted tree-decomposition $\mathcal{T}_C = (T_C, \mathcal{B}_C)$ of every connected component C of $G - X$ such that $|N_X(C)| \geqslant r$;

Repeat the following loop for every rooted tree-decomposition $\mathcal{T}_C$;
while $\mathcal{T}_C$ *contains an unprocessed bag* **do**
  Let B be an unprocessed bag at the farthest distance from the root of $\mathcal{T}_C$;
  [**LCA marking step**]
  if B *is the LCA of two marked bags of* $\mathcal{M}$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove the vertices of B from every bag of $\mathcal{T}_C$;
  [**Large-subgraph marking step**]
  else if G_B *contains a connected component* C_B *such that* $|N_X(C_B)| \geqslant r$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{B\}$ and remove the vertices of B from every bag of $\mathcal{T}_C$;
  Bag B is now processed;

return $Y_0 = X \cup V(\mathcal{M})$;

In other words, each cluster of $G - Y_0$ is a restricted $(2t+r)$-protrusion. Note that Proposition 1 neither bounds ℓ nor $|Y_0|$. In the sequel, we will use Algorithm 1 and Proposition 1 to give explicit bounds on ℓ and $|Y_0|$, in order to achieve our two main results.

4 Linear Kernels on Graphs Excluding a Topological Minor

In this section we prove our first main result (Theorem I). We then state a number of concrete problems that satisfy the structural constraints imposed by this theorem and discuss these constraints in the context of previous work in this area. With the protrusion machinery of Section 2 at hand, we can now describe the protrusion reduction rule. In the following, we will drop the subscript from the protrusion limit functions ρ_Π and ρ'_Π.

Reduction Rule 1 (Protrusion reduction rule). *Let $\Pi_\mathcal{G}$ denote a parameterized graph problem restricted to some graph class $\mathcal{G}$, let $(G,k) \in \Pi_\mathcal{G}$ be a* YES*-instance of $\Pi_\mathcal{G}$, and let $t \in \mathbb{N}$ be a constant. Suppose that $W' \subseteq V(G)$ is a t-protrusion of G such that $|W'| > \rho'(t)$. Let $W \subseteq V(G)$ be a $2t$-protrusion of G such that $\rho'(t) < |W| \leqslant 2 \cdot \rho'(t)$, obtained as described in Lemma 2. We let G_W denote the $2t$-boundaried graph $G[W]$ with boundary $\mathbf{bd}(G_W) = \partial_G(W)$. Let further $G_1 \in \mathcal{R}_{2t}$ be the representative of G_W for the equivalence relation $\equiv_{\Pi,|\partial(W)|}$ as defined in Lemma 1. The protrusion reduction rule (for boundary size t) is the following:* Reduce (G,k) to $(G',k') = (G \ominus G_W \oplus G_1, k - \Delta_{\Pi,2t}(G_1,G_W))$.

By Lemma 1, the parameter in the new instance does not increase. The safety of the above reduction rule is shown in the full version. Note that if (G,k) is reduced w.r.t. the protrusion reduction rule with boundary size β, then for all $t \leqslant \beta$, every t-protrusion W of G has size at most $\rho'(t)$.

Definition 1 (Treewidth-bounding). *A parameterized graph problem $\Pi_{\mathcal{G}}$ is called* (s,t)-treewidth-bounding *for a function $s\colon \mathbb{N} \to \mathbb{N}$ and a constant t if for all $(G,k) \in \Pi$ there exists $X \subseteq V(G)$ (the* treewidth-modulator*) such that $|X| \leqslant s(k)$ and $\mathbf{tw}(G - X) \leqslant t - 1$. We call $\Pi_{\mathcal{G}}$* treewidth-bounding on a graph class $\mathcal{G}$ *if this condition holds under the restriction that $G \in \mathcal{G}$. We call s the* treewidth-modulator size *and t the* treewidth bound *of the problem $\Pi_{\mathcal{G}}$.*

We assume in the following that the problem $\Pi_{\mathcal{G}}$ at hand is (s,t)-treewidth-bounding. Note that in general s,t depend on $\Pi_{\mathcal{G}}$ and $\mathcal{G}$.

We first prove a slight generalization of Theorem I which highlights all the key ingredients required. To this end, we define the *constriction* operation, which essentially shrinks paths into edges.

Definition 2 (Constriction). *Let G be a graph and let $\mathcal{P}$ be a set of paths in G such that for each $P \in \mathcal{P}$ we have* (1) *the endpoints of P are not connected by an edge in G; and* (2) *for all $P' \in \mathcal{P}$, with $P' \neq P$, $V(P) \cap V(P')$ has at most one vertex, which must also be an endpoint of both paths. We define the* constriction *of G under $\mathcal{P}$, denoted by $G|_{\mathcal{P}}$, as the graph H obtained by connecting the endpoints of each $P \in \mathcal{P}$ by an edge and then removing all inner vertices of P.*

We say that H is a *d-constriction* of G if there exists $G' \subseteq G$ and a set of paths $\mathcal{P}$ in G' such that $d = \max_{P \in \mathcal{P}} |P|$ and $H = G'|_{\mathcal{P}}$. Given graph classes $\mathcal{G}, \mathcal{H}$ and some integer $d \geqslant 2$, we say that $\mathcal{G}$ *d-constricts into $\mathcal{H}$* if for every $G \in \mathcal{G}$, every possible d-constriction H of G is contained in the class $\mathcal{H}$. For the case that $\mathcal{G} = \mathcal{H}$ we say that $\mathcal{G}$ is *closed under d-constrictions*. We will call $\mathcal{H}$ the *witness* class, as the proof of Theorem 1 works by taking an input graph G and constricting it into some witness graph H whose properties will yield the desired bound on $|G|$. We let $\omega(G)$ denote the size of a largest clique in G and $\#\omega(G)$ the total number of cliques in G (not necessarily maximal ones).

Theorem 1. [$\star$] *Let $\mathcal{G}, \mathcal{H}$ be graph classes closed under taking subgraphs such that $\mathcal{G}$ d-constricts into $\mathcal{H}$ for a fixed constant $d \in \mathbb{N}$. Assume that $\mathcal{H}$ has the property that there exist functions $f_E, f_{\#\omega}\colon \mathbb{N} \to \mathbb{N}$ and a constant $\omega_{\mathcal{H}}$ (depending only on $\mathcal{H}$) such that for each graph $H \in \mathcal{H}$ the following conditions hold:*

$$|E(H)| \leqslant f_E(|H|), \quad \#\omega(H) \leqslant f_{\#\omega}(|H|), \quad \text{and } \omega(H) < \omega_{\mathcal{H}}.$$

Let Π be a parameterized graph problem that has FII and is (s,t)-treewidth-bounding, both on the graph class $\mathcal{G}$. Define $x_k := s(k) + 2t \cdot f_E(s(k))$. Then any reduced instance $(G,k) \in \Pi$ has a protrusion decomposition $V(G) = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ such that: (1) $|Y_0| \leqslant x_k$; (2) $|Y_i| \leqslant \rho'(2t + \omega_{\mathcal{H}})$ for $1 \leqslant i \leqslant \ell$; and (3) $\ell \leqslant f_{\#\omega}(x_k) + x_k + 1$. Hence Π restricted to $\mathcal{G}$ admits kernels of size at most $x_k + (f_{\#\omega}(x_k) + x_k + 1)\rho'(2t + \omega_{\mathcal{H}})$.

Theorem 1 directly implies the following, using the fact that H-topological-minor-free graphs are ϵ-degenerate.

Theorem 2. [$\star$] *Fix a graph H and let $\mathcal{G}_H$ be the class of H-topological-minor-free graphs. Let Π be a parameterized graph-theoretic problem that has FII and is $(s_{\Pi,\mathcal{G}_H}, t_{\Pi,\mathcal{G}_H})$-treewidth-bounding on the class $\mathcal{G}_H$. Then Π admits a kernel of size $O(s_{\Pi,\mathcal{G}_H}(k))$.*

Theorem I is now just a consequence of the special case for which the treewidth-bound is linear. We present concrete problems that are affected by our result.

Corollary 1. *The following problems are linearly treewidth-bounding and have FII on $\mathcal{G}_H$ and hence admit linear kernels on $\mathcal{G}_H$:* Vertex Cover[4]*;* Cluster Vertex Deletion[4]*;* Feedback Vertex Set*;* Chordal Vertex Deletion*;* Interval *and* Proper Interval Vertex Deletion*;* Cograph Vertex Deletion*;* Edge Dominating Set*.*

Theorem I requires problems to be treewidth-bounding, at first glance, a quite strong restriction. However, the property of being treewidth-bounding appears implicitly or explicitly in previous work on linear kernels on sparse graphs [4,15].

5 Single-Exponential Algorithm for Planar-$\mathcal{F}$-Deletion

This section is devoted to the single-exponential algorithm for the Planar-$\mathcal{F}$-Deletion problem. Let henceforth H_p be some fixed (connected or disconnected) arbitrary planar graph in the family $\mathcal{F}$, and let $r := |V(H_p)|$. First of all, using iterative compression, we reduce the problem to obtaining a single-exponential algorithm for the Disjoint Planar-$\mathcal{F}$-Deletion problem, which is defined as follows: given a graph G and a subset of vertices $X \subseteq V(G)$ such that $G - X$ is H-minor-free for every $H \in \mathcal{F}$, compute a set $\tilde{X} \subseteq V(G)$ disjoint from X such that $|\tilde{X}| < |X|$ and $G - \tilde{X}$ is H-minor-free for every $H \in \mathcal{F}$, if such a set exists. The parameter is $k = |X|$.

The input set X is called the *initial solution* and the set $\tilde{X}$ the *alternative solution.* Let $t_{\mathcal{F}}$ be a constant (depending on the family $\mathcal{F}$) such that $\mathbf{tw}(G - X) \leqslant t_{\mathcal{F}} - 1$ (note that such a constant exists by Robertson and Seymour [23]). The following lemma relies on the fact that being $\mathcal{F}$-minor-free is a hereditary property with respect to induced subgraphs. For a proof, see for instance [6, 19–21].

Lemma 4. *If the parameterized* Disjoint Planar-$\mathcal{F}$-Deletion *problem can be solved in time $c^k \cdot p(n)$, where c is a constant and $p(n)$ is a polynomial in n, then the* Planar-$\mathcal{F}$-Deletion *problem can be solved in time $(c + 1)^k \cdot p(n) \cdot n$.*

To solve Disjoint Planar-$\mathcal{F}$-Deletion, we first construct a protrusion decomposition using Algorithm 1 with input (G, X, r). But it turns out that the set Y_0

[4] Listed for completeness; these problems have a kernel with a linear number of vertices on general graphs.

output by Algorithm 1 does not define a *linear* protrusion decomposition of G, which is crucial for our purposes. To circumvent this problem, our strategy is to guess the intersection I of the alternative solution $\tilde{X}$ with the set Y_0. As a result, we obtain Proposition 2, which is fundamental in order to prove Theorem II.

Proposition 2 (Linear protrusion decomposition). *Let (G, X, k) be a* Yes-*instance of the* Disjoint Planar-$\mathcal{F}$-Deletion *problem. There exists a $2^{O(k)} \cdot n$-time algorithm that identifies a set $I \subseteq V(G)$ of size at most k and a $(O(k), 2t_{\mathcal{F}} + r)$-protrusion decomposition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ of $G - I$ such that:* (1) *$X \subseteq Y_0$; and* (2) *there exists a set $X' \subseteq V(G) \setminus Y_0$ of size at most $k - |I|$ such that $G - \tilde{X}$, with $\tilde{X} = X' \cup I$, is H-minor-free for every graph $H \in \mathcal{F}$.*

Towards the proof of Proposition 2, we need the following ingredient.

Proposition 3 (Thomason [25], Fomin, Oum, and Thilikos [16]). *There is a constant $\alpha < 0.320$ such that every n-vertex graph with no K_r-minor has at most $(\alpha r \sqrt{\log r}) \cdot n$ edges. There is a constant $\mu < 11.355$ such that, for $r > 2$, every n-vertex graph with no K_r-minor has at most $2^{\mu r \log \log r} \cdot n$ cliques.*

For the sake of simplicity, let henceforth $\alpha_r := \alpha r \sqrt{\log r}$ and $\mu_r := 2^{\mu r \log \log r}$. For each guessed set $I \subseteq Y_0$, we denote $G_I := G - I$.

Lemma 5. [$\star$] *If (G, X, k) is a* Yes-*instance of the* Disjoint Planar-$\mathcal{F}$-Deletion *problem, then the set $Y_0 = V(\mathcal{M}) \cup X$ of vertices returned by Algorithm 1 has size at most $k + 2t_{\mathcal{F}} \cdot (1 + \alpha_r) \cdot k$.*

Lemma 6. [$\star$] *If $(G_I, Y_0 \setminus I, k - |I|)$ is a* Yes-*instance of the* Disjoint Planar-$\mathcal{F}$-Deletion *problem, then the number of clusters of $G_I - Y_0$ is at most $(5t_{\mathcal{F}} \alpha_r \mu_r) \cdot k$, where Y_0 is the set of vertices returned by Algorithm 1.*

We are now ready to prove Proposition 2.

Proof (of Proposition 2). By Lemma 5, we can compute in linear time a set Y_0 of $O(k)$ vertices containing X such that every cluster of $G - Y_0$ is a restricted $(2t_{\mathcal{F}} + r)$-protrusion. If (G, X, k) is a Yes-instance of the Disjoint Planar-$\mathcal{F}$-Deletion problem, then there exists a set $\tilde{X}$ of size at most $|X|$ and disjoint from X such that $G - \tilde{X}$ does not contain any graph $H \in \mathcal{F}$ as a minor. Branching on every possible subset of $Y_0 \setminus X$, one can guess the intersection I of $\tilde{X}$ with $Y_0 \setminus X$. By Lemma 5, the branching degree is $2^{O(k)}$. As (G, X, k) is a Yes-instance, for at least one of the guessed subsets I, the instance $(G_I, Y_0 \setminus I, k - |I|)$ is a Yes-instance of the Disjoint Planar-$\mathcal{F}$-Deletion problem. Now, by Lemma 6, the partition $\mathcal{P} = (Y_0 \setminus I) \uplus Y_1 \uplus \cdots \uplus Y_\ell$, where $\{Y_1, \ldots, Y_\ell\}$ is the set of clusters of $G_I - Y_0$, is an $(O(k), r + 2t_{\mathcal{F}})$-protrusion decomposition of G_I.

By Proposition 2, we can focus on solving Disjoint Planar-$\mathcal{F}$-Deletion in single-exponential time when a linear protrusion decomposition is given. To that aim, we define an equivalence relation on subsets of vertices of each restricted protrusion Y_i. The key observation is that each of these equivalence relations defines *finitely* many equivalence classes such that any partial solution lying on Y_i can be replaced with one of the representatives while preserving the feasibility. This basically follows from the *finite index* of MSO-definable properties (see, e.g., [5]). Then, we use a *decomposability* property of the solution, namely, that there always exists a solution which is formed by the union of one representative per restricted protrusion. Finally, in order to make the algorithm fully *constructive* and *uniform* on the family $\mathcal{F}$, we use classic arguments from tree automaton theory, such as the *method of test sets*. All details can be found in the full version.

Proposition 4. [$\star$] *Let* (G, Y_0, k) *be an instance of* Disjoint Planar-$\mathcal{F}$-Deletion *and let* $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ *be an* (α, β)*-protrusion decomposition of* G*, for some constant* β*. There exists an* $O(2^\ell \cdot n)$*-time algorithm which computes a solution* $\tilde{X} \subseteq V(G) \setminus Y_0$ *of size at most* k *if it exists, or correctly decides that there is no such solution.*

We finally have all the ingredients to piece everything together.

Proof (of Theorem II). Lemma 4 states that Planar-$\mathcal{F}$-Deletion can be reduced to Disjoint Planar-$\mathcal{F}$-Deletion so that the former is single-exponential time solvable provided that the latter is, and the degree of the polynomial function in n increases by one. We now proceed to solve Disjoint Planar-$\mathcal{F}$-Deletion in time $2^{O(k)} \cdot n$. Given an instance (G, X, k) of Disjoint Planar-$\mathcal{F}$-Deletion, we apply Proposition 2 to either correctly decide that (G, X, k) is a No-instance, or identify in time $2^{O(k)} \cdot n$ a set $I \subseteq V(G)$ of size at most k and a $(O(k), 2t_{\mathcal{F}} + r)$-protrusion decomposition $\mathcal{P} = Y_0 \uplus Y_1 \uplus \cdots \uplus Y_\ell$ of $G - I$, with $X \subseteq Y_0$, such that there exists a set $X' \subseteq V(G) \setminus Y_0$ of size at most $k - |I|$ such that $G - \tilde{X}$, with $\tilde{X} = X' \cup I$, is H-minor-free for every graph $H \in \mathcal{F}$. Finally, using Proposition 4 we can solve the instance $(G_I, Y_0 \setminus I, k - |I|)$ in time $2^{O(k)} \cdot n$.

Acknowledgements. We thank D. M. Thilikos, B. Courcelle, D. Lokshtanov, and S. Saurabh for interesting discussions and helpful remarks on the manuscript.

References

1. Alber, J., Fellows, M.R., Niedermeier, R.: Polynomial-time data reduction for Dominating Set. Journal of the ACM 51, 363–384 (2004)
2. Bodlaender, H.L.: Dynamic programming on graphs with bounded treewidth. In: Lepistö, T., Salomaa, A. (eds.) ICALP 1988. LNCS, vol. 317, pp. 105–118. Springer, Heidelberg (1988)
3. Bodlaender, H.L.: A linear time algorithm for finding tree-decompositions of small treewidth. SIAM Journal on Computing 25, 1305–1317 (1996)
4. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) Kernelization. In: Proc. of 50th FOCS, pp. 629–638. IEEE Computer Society (2009)
5. Bodlaender, H.L., van Antwerpen-de Fluiter, B.: Reduction algorithms for graphs of small treewidth. Information and Computation 167(2), 86–119 (2001)

6. Chen, J., Fomin, F.V., Liu, Y., Lu, S., Villanger, Y.: Improved algorithms for feedback vertex set problems. Journal of Computer and System Sciences 74(7), 1188–1198 (2008)
7. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: An improved FPT algorithm and quadratic kernel for pathwidth one vertex deletion. In: Raman, V., Saurabh, S. (eds.) IPEC 2010. LNCS, vol. 6478, pp. 95–106. Springer, Heidelberg (2010)
8. Dehne, F., Fellows, M., Langston, M.A., Rosamond, F., Stevens, K.: An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem. In: Wang, L. (ed.) COCOON 2005. LNCS, vol. 3595, pp. 859–869. Springer, Heidelberg (2005)
9. Diestel, R.: Graph Theory, 4th edn. Springer, Heidelberg (2010)
10. Dinneen, M.: Too many minor order obstructions. Journal of Universal Computer Science 3(11), 1199–1206 (1997)
11. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer (1999)
12. Fellows, M.R., Langston, M.A.: Nonconstructive tools for proving polynomial-time decidability. Journal of the ACM 35, 727–739 (1988)
13. Fomin, F.V., Lokshtanov, D., Misra, N., Philip, G., Saurabh, S.: Hitting forbidden minors: Approximation and kernelization. In: STACS 28th. LIPIcs, vol. 9, pp. 189–200. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2011)
14. Fomin, F.V., Lokshtanov, D., Misra, N., Saurabh, S.: Planar $\mathcal{F}$-Deletion: Approximation and Optimal FPT Algorithms. In: Proc. of 53rd FOCS, pp. 470–479. IEEE Computer Society (2012)
15. Fomin, F.V., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Bidimensionality and kernels. In: Proc. of 21st SODA, pp. 503–510. SIAM (2010)
16. Fomin, F.V., Oum, S., Thilikos, D.M.: Rank-width and tree-width of H-minor-free graphs. European Journal of Combinatorics 31(7), 1617–1628 (2010)
17. Guo, J., Gramm, J., Hüffner, F., Niedermeier, R., Wernicke, S.: Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. Journal of Computer and System Sciences 72(8), 1386–1396 (2006)
18. Guo, J., Niedermeier, R.: Linear problem kernels for NP-hard problems on planar graphs. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 375–386. Springer, Heidelberg (2007)
19. Joret, G., Paul, C., Sau, I., Saurabh, S., Thomassé, S.: Hitting and harvesting pumpkins. In: Demetrescu, C., Halldórsson, M.M. (eds.) ESA 2011. LNCS, vol. 6942, pp. 394–407. Springer, Heidelberg (2011)
20. Kim, E.J., Paul, C., Philip, G.: A single-exponential FPT-algorithm for K_4-minor cover problem. In: Fomin, F.V., Kaski, P. (eds.) SWAT 2012. LNCS, vol. 7357, pp. 119–130. Springer, Heidelberg (2012)
21. Lokshtanov, D., Saurabh, S., Sikdar, S.: Simpler parameterized algorithm for OCT. In: Fiala, J., Kratochvíl, J., Miller, M. (eds.) IWOCA 2009. LNCS, vol. 5874, pp. 380–384. Springer, Heidelberg (2009)
22. Philip, G., Raman, V., Villanger, Y.: A quartic kernel for Pathwidth-One Vertex Deletion. In: Thilikos, D.M. (ed.) WG 2010. LNCS, vol. 6410, pp. 196–207. Springer, Heidelberg (2010)
23. Robertson, N., Seymour, P.D.: Graph minors II. Algorithmic aspects of tree-width. Journal of Algorithms 7, 309–322 (1986)
24. Robertson, N., Seymour, P.D.: Graph minors XIII. The disjoint paths problem. Journal of Combinatorial Theory, Series B 63, 65–110 (1995)
25. Thomason, A.: The extremal function for complete minors. Journal of Combinatorial Theory, Series B 81(2), 318–338 (2001)