Linear Kernels for (Connected) Dominating Set on H-minor-free graphs

Fedor V. Fomin * Daniel Lokshtanov† Saket Saurabh‡ Dimitrios M. Thilikos§

Abstract

We give the first linear kernels for DOMINATING SET and CONNECTED DOMINATING SET problems on graphs excluding a fixed graph H as a minor. In other words, we give polynomial time algorithms that, for a given H-minor free graph G and positive integer k, output an H-minor free graph G' on $\mathcal{O}(k)$ vertices such that G has a (connected) dominating set of size k if and only if G' has. Prior to our work, the only polynomial kernel for DOMINATING SET on graphs excluding a fixed graph H as a minor was due to Alon and Gutner [ECCC 2008, IWPEC 2009] and to Philip, Raman, and Sikdar [ESA 2009] but the size of their kernel is $k^{c(H)}$, where $c(H)$ is a constant depending on the size of H. Alon and Gutner asked explicitly, whether one can obtain a linear kernel for DOMINATING SET on H-minor free graphs. We answer this question in affirmative. For CONNECTED DOMINATING SET no polynomial kernel on H-minor free graphs was known prior to our work.

Our results are based on a novel generic reduction rule producing an equivalent instance of the problem with treewidth $\mathcal{O}(\sqrt{k})$. The application of this rule in a divide-and-conquer fashion together with protrusion techniques brings us to linear kernels.

As a byproduct of our results we obtain the first subexponential time algorithms for CONNECTED DOMINATING SET, a deterministic algorithm solving the problem on an n-vertex H-minor free graph in time $2^{\mathcal{O}(\sqrt{k}\log k)} + n^{\mathcal{O}(1)}$ and a Monte Carlo algorithm of running time $2^{\mathcal{O}(\sqrt{k})} + n^{\mathcal{O}(1)}$. For DOMINATING SET our results implies a significant simplification and refinement of a $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$ algorithm on H minor free graphs due to Demaine et al. [SODA 2003, J. ACM 2005].

Keywords: Parameterized complexity, kernelization, algorithmic graph minors, dominating set, connected dominating set

*Department of Informatics, University of Bergen, Norway.

†University of California, San Diego, USA.

‡The Institute of Mathematical Sciences, CIT Campus, Chennai, India.

§Department of Mathematics, National & Kapodistrian University of Athens, Greece. Supported by the project "Kapodistrias" (AΠ 02839/28.07.2008) of the National and Kapodistrian University of Athens (project code: 70/4/8757).

1 Introduction

In the DOMINATING SET (DS) problem, we are given a graph G and a non-negative integer k, and the question is whether G contains a set of k vertices whose closed neighborhood contains all the vertices of G. In the connected variant, CONNECTED DOMINATING SET (CDS), we also demand the subgraph induced by the dominating set to be connected. DS, together with its numerous variants, is one of the most classic and well-studied problems in algorithms and combinatorics [33]. A considerable part of the algorithmic study on this NP-complete problem has been focused on the design of parameterized algorithms. Formally, a *parameterization* of a problem is assigning an integer k to each input instance and a parameterized problem is *fixed-parameter tractable* (FPT) if there is an algorithm that solves the problem in time $f(k) \cdot |I|^{O(1)}$, where $|I|$ is the size of the input and f is an arbitrary computable function depending on the parameter k only. In general, DS is W[2]-complete and therefore it cannot be solved by a parameterized algorithm, unless an unexpected collapse occurs in the Parameterized Complexity (see [22, 26, 40]). However, there are interesting graph classes where FPT-algorithms exist for the DS problem. The project of widening the horizon where such algorithms exist spanned a multitude of ideas that made DS the testbed for some of the most cutting-edge techniques of parameterized algorithm design. For example, the initial study of parameterized subexponential algorithms for DS on planar graphs [1, 15, 30] resulted in the creation of bidimensionality theory characterizing a broad range of graph problems that admit efficient approximate schemes or fixed-parameter solutions on a broad range of graphs [16, 18, 21].

Another emerging technique in parameterized complexity is *kernelization*. A parameterized problem is said to admit a *polynomial kernel* if there is a polynomial time algorithm (the degree of polynomial is independent of the parameter k), called a *kernelization* algorithm, that reduces the input instance down to an instance with size bounded by a polynomial $p(k)$ in k, while preserving the answer. This reduced instance is called a $p(k)$ *kernel* for the problem. If the size of the kernel is $O(k)$, then we call it a *linear kernel* (for a more formal definition, see Section 2). Kernelization appears to be an interesting both from practical and theoretical perspectives. Preprocessing (data reduction) is one of the basic practical algorithmic approaches. There are many real-world applications where even very simple preprocessing can be surprisingly effective, leading to significant size-reduction of the input. Kernelization is a natural tool not only for measuring the quality of preprocessing rules proposed for specific problems but also for designing new powerful preprocessing algorithms. From theoretical perspective, kernelization provides a deep insight into the hierarchy of parameterized problems in FPT, the most interesting class of parameterized problems. The recent breakthroughs [7, 14] establishes links between lower bounds on the sizes of kernels and classical computational complexity.

One of the first results on linear kernels is the celebrated work of Alber, Fellows, and Niedermeier on DS on planar graphs [2]. This work augmented significantly the interest in proving polynomial (or preferably linear) kernels for other parameterized problems. The result from [2], see also [9], has been extended to much more general graph classes like graphs of bounded genus [8] and apex-minor free graphs [29]. An important step in this direction was done by Alon and Gutner [3, 32] who obtained a kernel of size $O(k^h)$ for DS on H-minor free graphs, where the constant h depends on the excluded graph H. Later, Philip, Raman, and Sikdar [41] obtained a kernel of size $O(k^h)$ on $K_{i,j}$-free and d-degenerated graphs, where h depends on i, j and d respectively. Sizes of kernels in [3, 32, 41] are bounded by polynomials in k with degrees depending on the size of the excluded minor H. Therefore, the challenge is to ask for polynomial kernels of size $f(h) \cdot k^{O(1)}$, where the function f depends *exclusively* on the graph class. In this direction, there are already results for more restricted graph classes. According to the meta-algorithmic results on kernels introduced in [8], DS has a kernel of size $f(g) \cdot k$ on graphs of genus g. Recently, an alternative meta-algorithmic framework, based on bidimensionality theory [16], was introduced in [29], implying the existence of a kernel of size $f(H) \cdot k$ for DS on graphs excluding an apex graph H as a minor. While apex-minor free graphs form much more general class of graphs than graphs of bounded genus, H-minor free graphs form much larger class than apex-minor free graphs. For example, the class of graphs excluding $H = K_7$, the complete graph on 7 vertices, as a minor, contains all apex graphs. Alon and Gutner in [3] and Gutner in [32] posed as an open problem whether one can obtain a linear kernel for DS on H-minor free graphs.

In this work we obtain a linear kernel for DS on graphs excluding some fixed graph H as a minor, which answers affirmatively the question of Alon and Gutner. Moreover, a non-trivial modification of the ideas for DS kernelization can be used to obtain a linear kernel for CDS, which is usually much more difficult problem to handle due to connectivity constrains. The extension of the results for planar graphs from [2] and apex-minor free graphs from [29] to the more general family of H-minor free graphs cannot be straightforward. Similar difficulties in transition of algorithmic techniques from apex-minor free to H-minor free graphs were observed in approximation [19] and parameterized algorithms [16, 23]. Intuitively, the explanation is that excluding an apex graph makes it possible to bound the tree-decomposability of the input graph by a *sublinear* function of the size of a dominating set which is not the case for more general classes of H-minor free graphs.

The main idea behind our algorithm is to identify and remove "irrelevant" vertices without changing the solution such that in the reduced graph one can select $\mathcal{O}(k)$ vertices which removal leaves protrusions, that is, subgraphs of constant treewidth separated from the remaining vertices by a constant number of vertices. As far as we are able to obtain such a graph, we can use the techniques from [29] to construct the linear kernel. Roughly speaking, our rule to identify "irrelevant" vertices works as follows: we try specific vertex subsets of constant size, for each subset we try all "feasible" scenarios how dominating sets can interact with the subset, and find neighbours of theses subsets which removal does not change the outcome of any feasible scenario. The main difference of this new reduction rule in comparison to other rules for DS [2, 9] is that instead of reducing the size of the graph to $\mathcal{O}(k)$, it reduces the treewidth of the graph to $\mathcal{O}(\sqrt{k})$. Thus idea-wise, it is more closer to the "irrelevant vertex" approach developed by Robertson and Seymour for disjoint paths and minor checking problems [42]. However, the significant difference with this technique is that in all applications of "irrelevant vertex" the bounds on the treewidth are exponential or even worse [36, 37, 38]. Moreover, Adler et al. [34] provide instances of the disjoint paths problem on planar graphs, for which the irrelevant vertex approach of Robertson and Seymour produces graphs of treewidth $2^{\Omega(k)}$. Our rule provides a reduced graph with *sublinear* treewidth.

The proof that after deletion of all irrelevant vertices the treewidth of the graph becomes sublinear is non-trivial. For this proof we need the theorem of Robertson and Seymour [43] on decomposing a graph into a set of torsos connected via clique-sums. By making use of this theorem, we show that by applying the rule for all subsets of apex vertices of each torso, it is possible to reduce the treewidth of each torso to $\mathcal{O}(\sqrt{k})$. This implies that the treewidth of the reduced graph is also $\mathcal{O}(\sqrt{k})$. However, the number of torsos can be $\Omega(n)$, and the sublinear treewidth of the reduced graph still does not bring us directly to the desired constant-treewidth vertex removal property. To overcome this obstacle, we have to implement the irrelevant vertex rule in a divide and conquer manner, and only after that we can guarantee that the reduced graph has the constant-treewidth vertex removal property.

Besides linear kernels for DS and CDS, an immediate byproduct of our "irrelevant vertex technique" is a radical simplification of the subexponential parameterized algorithm of Demaine et al. [16] for DS, and the first parameterized subexponential algorithms for CDS on H-minor free graphs. Also our kernels can be used to obtain subexponential parameterized algorithms for these problems that use polynomial space.

2 Definitions and Notations

In this section we give various definitions which we make use of in the paper. We refer to Diestel's book [20] for standard definitions from Graph Theory. Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. A graph G' is a *subgraph* of G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. For subset $V' \subseteq V(G)$, the subgraph $G' = G[V']$ of G is called a *subgraph induced by* V' if $E(G') = \{uv \in E(G) \mid u, v \in V'\}$. By $N_G(u)$ we denote (open) neighborhood of u in graph G that is the set of all vertices adjacent to u and by $N[u] = N(u) \cup \{u\}$. Similarly, for a subset $D \subseteq V$, we define $N_G[D] = \cup_{v \in D} N_G[v]$ and $N_G(D) = N_G[D] \setminus D$. We omit the subscripts when it is clear from the context. Throughout the paper, given a graph G and vertex subsets Z and S, whenever we say that a subset Z *dominates all but (everything but)* S then we mean that $V(G) \setminus S \subseteq N[Z]$. Observe that a vertex of S can also be dominated by the set Z.

We denote by K_h the complete graph on h vertices. For integer $r \geq 1$ and vertex subsets $P, Q \subseteq V(G)$, we say that a subset Q is *r-dominated* by P, if for every $v \in Q$ there is $u \in P$ such that the distance between u and v is at most r. For $r = 1$, we simply say that Q is dominated by P. We denote by $N_G^r(P)$

the set of vertices r-dominated by P.

Given an edge $e = xy$ of a graph G, the graph G/e is obtained from G by contracting the edge e, that is, the endpoints x and y are replaced by a new vertex v_{xy} which is adjacent to the old neighbors of x and y (except from x and y). A graph H obtained by a sequence of edge-contractions is said to be a *contraction* of G. We denote it by $H \leq_c G$. A graph H is a *minor* of a graph G if H is the contraction of some subgraph of G and we denote it by $H \leq_m G$. We say that a graph G is H*-minor-free* when it does not contain H as a minor. We also say that a graph class $\mathcal{G}_H$ is H*-minor-free* (or, excludes H as a minor) when all its members are H-minor-free. An *apex graph* is a graph obtained from a planar graph G by adding a vertex and making it adjacent to some of the vertices of G. A graph class $\mathcal{G}_H$ is *apex-minor-free* if $\mathcal{G}_H$ excludes a fixed apex graph H as a minor. We denote by $\mathbf{tw}(G)$ the treewidth of graph G. (See Appendix for the definition of treewidth.)

Kernels and Protrusions. 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. We will assume that k is *given in unary* and hence $k \leq |x|^{\mathcal{O}(1)}$. 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 [22]. The notion of *kernelization* is formally defined as follows.

A *kernelization algorithm*, or in short, a *kernelization*, for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}$ is an algorithm that given $(x, k) \in \Gamma^* \times \mathbb{N}$ outputs in time polynomial in $|x| + k$ a pair $(x', k') \in \Gamma^* \times \mathbb{N}$ 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 of kernelization $(x', k') \in \Pi$ is referred as the *kernel* and the function g is referred to as the size of the kernel. If $g(k) = k^{\mathcal{O}(1)}$ or $g(k) = \mathcal{O}(k)$ then we say that Π admits a polynomial kernel and linear kernel respectively.

Given a graph G, we say that a set $X \subseteq V(G)$ is an r*-protrusion* of G if $\mathbf{tw}(G[X]) \leq r$ and the number of vertices in X with a neighbor in $V(G) \setminus X$ is at most r.

Treewidth, torso and Graph Structure Theorem. A *tree decomposition* of a graph G is a pair $(\mathcal{X}, T)$ where T is a tree and $\mathcal{X} = \{X_i \mid i \in V(T)\}$ is a collection of subsets of V such that:
1. $\bigcup_{i \in V(T)} X_i = V(G)$,
2. for each edge $xy \in E(G)$, $\{x, y\} \subseteq X_i$ for some $i \in V(T)$;
3. for each $x \in V(G)$ the set $\{i \mid x \in X_i\}$ induces a connected subtree of T.

The *width* of the tree decomposition is $\max_{i \in V(T)} |X_i| - 1$. The *treewidth* of a graph G is the minimum width over all tree decompositions of G. We denote by $\mathbf{tw}(G)$ the treewidth of graph G.

A *torso* of a tree-decomposition $(\mathcal{X}, T)$ of a graph G is a graph L_t, $t \in V(T)$, obtained from $G[X_t]$ by adding edges uv such that u and v are in $X_t \cap X_{t'}$, where t and t' are nodes adjacent in T. Observe that it is possible that u and v may not be adjacent in G and thus L_t is not necessarily a subgraph of G. To state the next theorem we need also the notion of a graph that can be h-nearly embedded in a surface.

Definition 1 (h-nearly embeddable graphs)**.** *Let Σ be a surface with boundary cycles $C_1, \ldots, C_h$, i.e. each cycle C_i is the border of a disc in Σ. A graph G is h-*nearly embeddable *in Σ, if G has a subset X of size at most h, called* apices*, such that there are (possibly empty) subgraphs $G_0 = (V_0, E_0), \ldots, G_h = (V_h, E_h)$ of $G \setminus X$ such that*

- *$G \setminus X = G_0 \cup \cdots \cup G_h$,*
- *G_0 is embeddable in Σ, we fix an embedding of G_0,*
- *graphs $G_1, \ldots, G_h$ (called* vortices*) are pairwise disjoint,*
- *for $1 \leq \cdots \leq h$, let $U_i := \{u_{i_1}, \ldots, u_{i_{m_i}}\} = V_0 \cap V_i$, G_i has a path decomposition (B_{ij}), $1 \leq j \leq m_i$, of width at most h such that*
  - *for $1 \leq i \leq h$ and for $1 \leq j \leq m_i$ we have $u_j \in B_{ij}$*
  - *for $1 \leq i \leq h$, we have $V_0 \cap C_i = \{u_{i_1}, \ldots, u_{i_{m_i}}\}$ and the points $u_{i_1}, \ldots, u_{i_{m_i}}$ appear on C_i in this order (either if we walk clockwise or anti-clockwise).*

The following theorem is one of the most fundamental results in Graph Minors Theory of Robertson and Seymour, see also Section 12.4 in Diestel's book [20].

Theorem 1 ([43])**.** *For every graph H there exists an integer h, depending only on the size of H, such that every graph excluding H as a minor has a tree-decomposition whose torsos can be h-nearly embedded in a surface Σ in which H cannot be embedded.*

Kawarabayashi and Wollan [35] prove a constructive version of this theorem, and gave an algorithm computing such a tree-decomposition of an n-vertex graph G in time $\mathcal{O}(f(H)n^3)$. Thus, throughout the paper we will assume that such a decomposition of the graph is given. So when we talk of the torsos of G we mean the torsos of this decomposition.

The main consequence of Theorem 1 we need for our purposes is that for every H there exist constants h and h' such that for every torso L of the decomposition from Theorem 1, there exists a set of vertices $A \subseteq V(L)$ of size at most h, called apices, such that the graph obtained from L after deleting the apices does not contain some apex graph H' of size h' as a minor. See, e.g. [31, Theorem 13].

3 Kernel for DOMINATING SET

In this section we give a linear kernel for the `DS` problem. The kernelization algorithm has two phases. The key ingredient of the first phase is a rule that removes "irrelevant vertices" in order to obtain an equivalent graph with treewidth bounded by $\mathcal{O}(\sqrt{k})$. The irrelevant vertex rule is applied in multiple rounds in a recursive fashion, and obtain a set D of size $\mathcal{O}(k)$ vertices such that its deletion leaves a graph of constant treewidth. We call such set D as *treewidth deletion set*. Then applying a "protrusion rule" [8] together with the fact that `DS` has finite integer index, we get the desired linear kernel for `DS`.

Obtaining an equivalent graph of treewidth at most $\mathcal{O}(\sqrt{k})$. Let G be an n-vertex graph excluding some fixed graph H as a minor. In this section we assume that we are given a tree-decomposition $(\mathcal{X}, T)$ of G as in Theorem 1, such that the torsos of the tree-decomposition can be h-nearly embedded in a surface Σ in which H cannot be embedded. Such a decomposition can be constructed in time $\mathcal{O}(f(H)n^3)$ [35]. Let L_t be a torso corresponding to some vertex $t \in V(T)$. Next we show how to obtain an equivalent graph G', in fact an induced subgraph of G obtained by deleting vertices from L_t, such that the subgraph corresponding to L_t in G' has treewidth $\mathcal{O}(\sqrt{k})$. We repeat this procedure for every torso corresponding to vertices in $V(T)$. Finally we obtain an equivalent graph G' such that it has a tree-decomposition $(\mathcal{X}', T)$ such that all its torsos have treewidth $\mathcal{O}(\sqrt{k})$. Since the treewidth of a graph is at most the maximum treewidth of its torsos, see e.g. [16], this implies that the treewidth of G' is $\mathcal{O}(\sqrt{k})$.

Irrelevant Vertex Rule. The irrelevant vertex rule will be used in a recursive fashion. In each recursive step it is used in order to reduce the treewidth of torsos and hence also the entire graph. Then the graph is split in two pieces and the procedure is applied recursively to the two pieces. In this way we obtain a treewidth deletion set of the reduced graph of size $\mathcal{O}(k)$. Let G be a graph given with its tree-decomposition $(\mathcal{X}, T)$ as described in Theorem 1, and L_t be one of its torsos. Let S be a dominating set of G, and A, $|A| \leq h$, be the set of apices of L_t. The reduction rule essentially "preserves" all dominating sets of size at most $|S|$ in G, without introducing any new ones.

To describe the reduction rule we need several definitions. The first step in our reduction rule is to classify different subsets A' of A into feasible and infeasible sets. The intuition behind the definition is that a subset A' of A is feasible if there exists a set D in G of size at most $|S| + 1$ that dominates all but S and so that $D \cap A = A'$. However, we cannot test in polynomial time whether such a set D exists. We will therefore say that a subset A' of A is *feasible* if the 2-approximation for Dominating Set on H-minor-free graphs [17, 28] outputs a set D of size at most $2(|S| + 2)$ that dominates $V(G) \setminus (A \cup S)$. Observe that if such a set D of size at most $|S| + 1$ exists then A' is surely feasible, while if no such set D of size at most $2|S| + 2$ exists, then A' is surely not feasible. We will frequently use this in our arguments. Let us remark that there always exists a feasible set $A' \subseteq A$. In particular, $A' = S \cap A$ is feasible since S dominates G. For feasible sets A' we will denote by $D(A')$ the set D output by the approximation algorithm.

For every subset $A' \subseteq A$, we select a vertex v of G such that $A' \subseteq N_G[v]$. If such a vertex exist, we call it a *representative* of A'. Let us remark that some sets can have no representatives and some distinct subsets of A may have the same representative. We define R to be the set of representative vertices for subsets of A. The size of R is at most $2^{|A|}$. For $A' \subseteq A$, the set of *dominated vertices* (by A') is $W(A') = N(A') \setminus A$. We say that vertex $v \in V(G) \setminus A$ is *fully dominated* by A' if $N[v] \setminus A \subseteq W(A')$. A vertex $w \in V(G) \setminus A$ is *irrelevant with respect to* A' if $w \notin R$, $w \notin S$, and w is fully dominated by A'. Now we are ready to state the irrelevant vertex rule.

Irrelevant Vertex Rule: If a vertex w is irrelevant with respect to every feasible $A' \subseteq A$, then delete w from G.

In the proof of correctness of the Irrelevant Vertex Rule, we will in fact prove that some stronger properties of the reduced graph hold than just that the reduced instance is equivalent to the instance we start from. The reason we do this is due to the recursive fashion in which the rule will be applied. To this end we need the following definition.

Definition 2. *Let G be a graph and S be a dominating set of G, G' be an induced subgraph of G, and $S' \subseteq V(G')$ be a dominating set of G' such that $S \subseteq S'$. We say that (G', S')* simulates *(G, S) if the following two conditions are satisfied. (i) For every set Z in G of size at most $|S|$ that dominates $V(G) \setminus S$, there is a set Z' of size at most $|Z|$ in G' such that $Z \cap S \subseteq Z'$ and Z' dominates all the vertices of $N_G[Z] \cap V(G')$; (ii) for every set Z' in G' that dominates all but S', there is a set Z in G of size at most $|Z'|$ such that $Z' \cap S' \subseteq Z$ and Z dominates $N_G[Z'] \cup (V(G) \setminus V(G'))$ in G.*

To get some intuition about the definition above, observe that if (G', S') simulates (G, S) then for every $p \leq |S|$, G has a dominating set of size p if and only if G' has a dominating set of size p. Given a dominating set of size p, say Z, of G, we know that there exists a set Z' of the same size that dominates all the vertices of $N[Z]$ present in G'. Since G' is an induced subgraph of G and Z is a dominating set, we have that $N[Z] \cap V(G') = V(G')$ and hence Z' is a dominating set of size at most p for G'. Similarly given a dominating set Z' of G' of size at most p, we know that there exists a set Z in G of size at most p that dominates $N[Z'] \cup (V(G) \setminus V(G')) = V(G)$.

There are two simple properties of simulation that we will frequently use. The first is that if $S \subset S'$ then (G, S') simulates (G, S). The second is transitivity: if (G'', S'') simulates (G', S') and (G', S') simulates (G, S), then (G'', S'') simulates (G, S). We are now ready to prove a lemma which will be crucial to arguing correctness of our kernelization algorithm.

Lemma 1. *Let S be a dominating set in G, and G' be the graph obtained by applying the Irrelevant Vertex Rule on G, where w was the deleted vertex. Then (G', S) simulates (G, S).*

Proof. We start with property (i) of simulation. Let $Z \subset V(G)$ such that $|Z| \leq |S|$ and Z dominates everything but S. Let $A' = Z \cap A$, and observe that A' is feasible because Z dominates all but S. If $w \notin Z$, then $Z' = Z$ satisfies condition (i), as $S \cap Z' = S \cap Z$, $|N(Z')| = |N(Z)|$, and $|Z'| = |Z|$. So assume $w \in Z$. Since w is irrelevant with respect to all feasible subsets of A and A' is feasible, we have that w is irrelevant with respect to A'. Hence $N_G(w) \setminus N_G(Z \setminus w) \subseteq A$. There is a representative $w' \in R$, $w' \neq w$ (since $w \notin R$), such that $N_G(w) \cap A \subseteq N_G(w') \cap A$. Hence $Z' = (Z \cup \{w'\}) \setminus \{w\}$ satisfies condition (i) of simulation.

Now, let $Z' \subseteq V(G')$ be such that $|Z'| \leq |S|$ and Z' dominates everything but S in G'. We show that Z' also dominates w in G. Specifically $Z' \cup \{w\}$ is a dominating set of all but S in G of size at most $|S| + 1$ so $Z' \cap A$ is feasible. Since $\{w\}$ is irrelevant with respect to $Z' \cap A$, we have $w \in N_G(Z' \cap A)$. This concludes the proof. □

Reducing treewidth. For a graph G and its dominating set S, we apply the Irrelevant Vertex Rule exhaustively on all torsos of G, obtaining an induced subgraph G' of G. By Lemma 1 and transitivity of simulation, (G', S) simulates (G, S). We now prove that a graph G which can not be reduced by the irrelevant vertex rule has low treewidth. The way we do this, is by proving that each torso of G has low treewidth.

Lemma 2. *Let G be a graph which is irreducible by the Irrelevant Vertex Rule and S be a dominating set of G. For every torso L_t of G, $\mathbf{tw}(L_t) = \mathcal{O}(\sqrt{|S|})$.*

Proof. Let $L_t^* = L_t \setminus A$, where A are the apices of L_t. We will obtain a 2-dominating set of size $\mathcal{O}(|S|)$ in L_t^*. Towards this end, consider the following set, $Q = \bigcup_{A' \subseteq A, A' \text{is feasible}} D(A') \cup R \cup S \setminus A$. The number of representative vertices R and the number of feasible subsets A' is at most $2^{|A|} \leq 2^h$, where h is a constant depending only on H. The size of $D(A')$ is at most $2|S| + 2$ for every A'. Thus $|Q| \leq 2^h(2|S|+2) + 2^h + |S| = \mathcal{O}(|S|)$. We prove that Q is a 2-dominating set of $V(G) \setminus A$. Let $w \in V(G) \setminus A$. If $w \in R$ or $w \in S$ then Q dominates S. So suppose $w \notin R \cup S$. Then, since w is not irrelevant, we have that there is a feasible subset A' of A such that w is relevant with respect to A'. Hence w is not fully dominated by A' and so w has a neighbour $w' \in V(G) \setminus N[A']$. But w' is dominated by $D(A') \subseteq Q$, and thus w is 2-dominated by Q in $G \setminus A$. Hence $G \setminus A$ has a 2-dominating set of size $\mathcal{O}(|S|)$.

The graph L_t^* can be obtained from $G \setminus A$ by contracting all edges in $E(G \setminus A) \setminus E(L_t^*)$ and adding all edges in $E(L_t^*) \setminus E(G \setminus A)$. Since contracting and adding edges does not increase the size of a minimum 2-dominating set of a graph, L_t^* has a 2-dominating set of size $\mathcal{O}(|S|)$.

To conclude, L_t^* excludes an apex graph as a minor (see discussions after Theorem 1) and it has a 2-dominating set of size $\mathcal{O}(|S|)$. By the bidimensionality of 2-dominating set, we have that $\mathbf{tw}(L_t^*) = \mathcal{O}(\sqrt{|S|})$ [16, 27]. Now we add all the apices of A to all the bags of the tree decomposition of L_t^* to obtain a tree decomposition for L_t' of width $\mathcal{O}(\sqrt{|S|}) + h = \mathcal{O}(\sqrt{|S|})$. □

Let us remark that Irrelevant Vertex Rule is based on the performance of a polynomial time approximation algorithm. Thus by Lemmata 1 and 2, and the fact that the treewidth of a graph is at most the maximum treewidth of its torsos, see e.g.[16], we obtain the following lemma.

Lemma 3. *There is a polynomial time algorithm that for a given graph G and a dominating set S of G, outputs an induced subgraph G' of G such that (G', S) simulates (G, S) and $\mathbf{tw}(G') = \mathcal{O}(\sqrt{|S|})$.*

Before we proceed further, we show the power of Lemma 3 by deriving a simple subexponential time algorithm for DS on H-minor free graphs. This is one of the cornerstone results in [16] and is based on a non-trivial two-layer dynamic programming over clique-sum decomposition tree of a H-minor free graphs. Lemma 3 can be used to obtain much simpler algorithm. Given a graph G and a positive integer k we first apply a factor 2-approximation algorithm given in [17, 28] for DS on G and obtain a set S. If the size of S is more than $2k$ then we return that G does not have a dominating set of size at most k. Otherwise, we apply Lemma 3 and obtain an equivalent graph G' such that $\mathbf{tw}(G') = \mathcal{O}(\sqrt{k})$. Now applying a constant factor approximation algorithm developed in [16] for computing the treewidth on G' we get a tree decomposition of width $\mathcal{O}(\sqrt{k})$. It is well known that checking whether a graph with treewidth t has a dominating set of size at most k can be done in time $\mathcal{O}(3^t n^{\mathcal{O}(1)})$ [44]. This together with the above bound on the treewidth, gives us an alternative proof of the following theorem.

Theorem 2 ([17])**.** *Given an n-vertex graph G excluding a fixed graph H as a minor, one can check whether G has a dominating set of size at most k in time $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$.*

Finding an equivalent graph with a treewidth deletion set of size $\mathcal{O}(k)$. Now we apply Lemma 3 recursively to obtain an $\mathcal{O}(k)$-sized treewidth deletion set. That is, given a graph G excluding a fixed graph H as a minor and a positive integer k, in polynomial time we output a graph G' such that (a) G has a dominating set of size at most k if and only if G' has a dominating set of size at most k; and (b) it is possible to remove $\mathcal{O}(k)$ vertices from G' such that the resulting graph is of constant treewidth. We need the following well known lemma, see e.g. [6], on separators in graphs of bounded treewidth.

Lemma 4. *Let G be a graph given with a tree-decomposition of width at most t and $w : V(G) \to \{0, 1\}$ be a weight function. Then in polynomial time we can find a bag X of the given tree-decomposition such that for every connected component $G[C]$ of $G \setminus X$, $w(C) \leq w(V(G))/2$. Furthermore, the connected components $C_1, \ldots, C_\ell$ of $G \setminus X$ can be grouped into two sets V_1 and V_2 such that $\frac{w(V(G))-w(X)}{3} \leq V_i \leq \frac{2(w(V(G))-w(X))}{3}$, for $i \in \{1, 2\}$.*

We proceed as follows. Given a graph G and a positive integer k, we first apply a factor 2-approximation algorithm from [17, 28] for DS on G and obtain a set S. If the size of S is more than $2k$ then we return that G does not have a dominating set of size at most k. If $|S| \leq 2k$ we will run a recursive procedure that will compute a new graph G^* and vertex sets S^* and D^* in G^* such that (G^*, S^*) simulates (G, S) and $\mathbf{tw}(G^* \setminus D^*) = \mathcal{O}(1)$. The set S will be passed down in recursive steps, so the recursive steps do not recalculate S. In particular we prove the following lemma.

Lemma 5. *Let G be an H-minor free graph and S be a dominating set of G. There is a polynomial time algorithm that computes an induced subgraph G^* of G, a dominating set S^* of G^*, and a set D^* such that (G^*, S^*) simulates (G, S), $\mathbf{tw}(G^* \setminus D^*) = \mathcal{O}(1)$ and $|D| = \mathcal{O}(|S|)$.*

Proof. By Lemma 3, we may assume that $\mathbf{tw}(G) = \mathcal{O}(\sqrt{|S|})$. Hence, if $|S| = \mathcal{O}(1)$ we are done, as we can return (G, S) with $D^* = \emptyset$. Otherwise, using a constant factor approximation of treewidth on H-minor-free graphs [25], we compute a tree-decomposition of G of width $d\sqrt{k}$. Now, by applying Lemma 4 on this decomposition, we find a partitioning of $V(G)$ into V_1, V_2 and X such that there are no edges from V_1 to V_2, $|X| \leq d\sqrt{k} + 1$, and $|V_i \cap S| \leq 2|S|/3$ for $i \in \{1, 2\}$. Let $S' = S \cup X$, and observe that (G, S') simulates (G, S).

We now apply the algorithm recursively on $(G[V_1 \cup X], S' \cap (V_1 \cup X))$ and $(G[V_2 \cup X], S' \cap (V_2 \cup X))$ and obtain graphs G_1, G_2 and sets S_1, S_2, D_1, D_2 such that for $i \in \{1, 2\}$

- (G_i, S_i) simulates $(G[V_i \cup X], S' \cap (V_i \cup X))$;
- $\mathbf{tw}(G_i \setminus D_i) = \mathcal{O}(1)$.

Since $X \subseteq S'$, we have that $S' \cap (V_i \cup X)$ is a dominating set of $G[V_i \cup X]$ and hence we actually can run the algorithm recursively on the two subcases. The algorithm will return $G^* = G[V(G_1) \cup V(G_2)]$ and the sets $S^* = S_1 \cup S_2$ and $D^* = D_1 \cup D_2 \cup X$. Clearly $\mathbf{tw}(G^* \setminus D^*) = \mathcal{O}(1)$, and we only need to prove that (G^*, S^*) simulates (G, S') and that $|D^*| = \mathcal{O}(|S|)$.

For (i) Let Z dominate all but S' in G, and $|Z| \leq 3|S'|$. For $i \in \{1, 2\}$ set $Z_i = Z \cap (V_i \cup X)$. We also assume that $|Z_i| \leq 3|S' \cap (V_i \cup X)|$. This need not be true in general, and in the case when $|Z_i| > 3|S' \cap (V_i \cup X)|$, we will take $Z_i = S' \cap (V_i \cup X)$. Now we prove for the case when it holds than $|Z_i| \leq 3|S' \cap (V_i \cup X)|$ for $i \in \{1, 2\}$, other cases are analogous.

Claim 1. *(G^*, S^*) simulates (G, S')*

Proof. It follows directly from the construction that $S' \subseteq S^*$, so it suffices to prove conditions (i) and (ii) of simulation. For (i), let Z dominate all but S' in G, and $|Z| \leq |S'|$. We set $Z_i = Z \cap (V_i \cup X)$, $i \in \{1, 2\}$. We also assume that $|Z_i| \leq |S' \cap (V_i \cup X)|$. This need not be true in general, and in the case when $|Z_i| > |S' \cap (V_i \cup X)|$, we will take $Z_i = S' \cap (V_i \cup X)$. Now we prove for the case when it holds than $|Z_i| \leq |S' \cap (V_i \cup X)|$ for $i \in \{1, 2\}$, other cases are analogous.

Since (G_i, S_i) simulates $(G[V_i \cup X], S' \cap (V_i \cup X))$, it follows that there exist sets Z'_1 and Z'_2 such that for $i \in \{1, 2\}$, Z'_i dominates $N_G[Z_i] \cap V(G_i)$ in G_i, $|Z'_i| \leq |Z_i|$, and $Z_i \cap S' \subseteq Z'_i$. We set $Z' = Z'_1 \cup Z'_2$, then

$$(Z \cap S) = (Z_1 \cap S) \cup (Z_2 \cap S) \subseteq (Z_1 \cap S_1) \cup (Z_2 \cap S_2) \subseteq Z'_1 \cup Z'_2 \subseteq Z'.$$

Furthermore, Z' dominates $N_G[Z] \cap (V(G_1) \cup V(G_2)) = N_G[Z] \cap V(G^*)$ because $N[Z] = N[Z_1] \cup N[Z_2]$, Z'_1 dominates $N[Z_1]$ and Z'_2 dominates $N[Z_2]$. Finally, $|Z'| \leq |Z|$ for the reason that

$$\begin{aligned}
|Z'| = |Z'_1 \cup Z'_2| &= |Z'_1| + |Z'_2| - |Z'_1 \cap Z'_2| \\
&\leq |Z_1| + |Z_2| - |Z'_1 \cap Z'_2| \\
&= |Z_1 \cup Z_2| + |Z_1 \cap Z_2| - |Z'_1 \cap Z'_2| \\
&= |Z_1 \cup Z_2| + |Z \cap X| - |Z'_1 \cap Z'_2| \\
&\leq |Z_1 \cup Z_2| = |Z|.
\end{aligned}$$

Now we prove that condition (ii) of simulation holds. We show that for every set Z' in G^* that dominates everything but S^*, there exists Z in G of size at most $|Z'|$, containing $Z' \cap S^*$, and dominating $N[Z']$ and the set of vertices in $V(G) \setminus V(G^*)$. For $i \in \{1, 2\}$, let $Z'_i = Z' \cap V_i$. Since (G_i, S_i) simulates $(G[V_i \cup X], S' \cap (V_i \cup X))$, there is a set $Z_i \subseteq V_i \cup X$ of size at most $|Z'_i|$ that dominates $N[Z'_i]$ and all of

$V_i \setminus V(G_i)$, and satisfies $Z'_i \cap S_i \subseteq Z_i$. Set $Z = Z_1 \cup Z_2$. By arguments identical to those in the forward direction, it follows that Z dominates $N_{G^*}[Z']$ and $V(G) \setminus V(G^*)$, that $Z' \cap S^* \subseteq Z$ and that $|Z| \leq |Z'|$. This concludes the proof of the claim. □

Finally, we upper bound $|D^*|$ by the following recursive formula.

$$|D^*| \;\leq\; \max_{1/3 \leq \alpha \leq 2/3} \left\{ \mu\left(\alpha|S| + d\sqrt{|S|}\right) + \mu\left((1-\alpha)|S| + d\sqrt{|S|}\right) + d\sqrt{|S|} \right\}.$$

Using simple induction one can show that the above solves to $\mathcal{O}(|S|)$. See for an example [28, Lemma 2]. Hence we conclude that $|D| = \mathcal{O}(|S|) = \mathcal{O}(k)$. This completes the proof of the lemma. □

Lemma 5 establishes that the output graph G^* and set S^* simulates (G, S). Since simulation implies that for every $p \leq |S|$, G^* has a dominating set of size p if and only if G does, and the set S was obtained by running a 2-approximation on G, we get the following lemma.

Lemma 6. *Let G be an H-minor free graph and S be a dominating set of G. There is a polynomial time algorithm that computes an induced subgraph G^* and a set D^* of size $\mathcal{O}(k)$ such that $\mathbf{tw}(G^* \setminus D^*) = \mathcal{O}(1)$ and G^* has a dominating set of size k if and only if G does.*

Final Kernel. Now we proceed to the proof of our main result for DS on graphs excluding a fixed graph H as a minor. We need the following lemmata.

Lemma 7 ([29, Lemma 3.4])**.** *For every fixed graph H and constant t there are constants ζ and r that satisfy the following. For any n-vertex graph G which excludes H as a minor and has a vertex set D^* of size k' such that $\mathbf{tw}(G \setminus D^*) \leq t$, then G has an r-protrusion of size at least $\zeta n/k'$.*

DS has finite integer index, and the following lemma is a special case of [29, Lemma 4.1], see also [8].

Lemma 8 ([8, 29])**.** *Let $\mathcal{G}_H$ be a class of graphs excluding a fixed graph H as a minor. Then there exists a constant c_r and an algorithm that given a graph $G \in \mathcal{G}_H$, an integer k and an r-protrusion X in G with $|X| > c_r$, runs in time $\mathcal{O}(|X|)$ and returns a graph $G^* \in \mathcal{G}_H$ and an integer k^* such that $|V(G^*)| < |V(G)|$, $k^* \leq k$, and G^* has a dominating set of size at most k^* if and only if G has a dominating set of size at most k.*

Theorem 3. *Let $\mathcal{G}_H$ be the class of graphs excluding a fixed graph H as a minor. Then DS has a linear kernel on $\mathcal{G}_H$.*

Proof. Given a graph G and a positive integer k, we apply Lemma 6 on G and S and obtain a graph G' such that G has a dominating set of size at most k if and only if G' has a dominating set of size at most k. We also obtain a treewidth deletion set D of size at most $\mathcal{O}(k)$, that is $\mathbf{tw}(G' \setminus D) \leq t$ for some fixed constant t.

By Lemma 7, G' contains an r-protrusion of size at least $\zeta|V(G')|/tk$. The reduction algorithm exhaustively applies Lemma 8. Since an irreducible instance contains no r-protrusion of size at least c_r, it follows that an irreducible instance (G', k') of DS must satisfy $\zeta|V(G')|/tk' < c_r$. Thus $|V(G')|$ is at most $k' \cdot tc_r/\zeta = \mathcal{O}(k)$.

The kernelization procedure runs in polynomial time because we can find a protrusion by guessing the boundary, which has constant size. Once given a protrusion X, we can replace it with an equivalent instance in $\mathcal{O}(|X|)$ time using the Lemma 8. This concludes the proof. □

The algorithm of Demaine et al. [17] computing a dominating set of size k in an n-vertex H-minor free graph uses exponential (in k) space $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$. Theorem 3 implies almost directly the following refinement of Theorem 2.

Theorem 4. *Given an n-vertex graph G excluding a fixed graph H as a minor, one can check whether G has a dominating set of size at most k in time $2^{\mathcal{O}(\sqrt{k})} + n^{\mathcal{O}(1)}$ and space $(nk)^{\mathcal{O}(1)}$.*

Proof. Our algorithm first applies Theorem 3 to obtain a graph with $O(k)$ vertices. Now we are assuming that the number of vertices in G is $n = \mathcal{O}(k)$. We solve a slightly more general version of domination, where we are given a subset S and the requirement is to find a set D of size at most k such that for every $v \in V(G) \setminus S$, $N[v] \cap D \neq \emptyset$. When $S = \emptyset$, the set D is a dominating set of size k. By the separator theorem of Alon et al. [4] for H-minor free graphs, one can find in polynomial time a partition of $V(G)$ into V_1, V_2 and X such that $|X| \leq \mathcal{O}(\sqrt{n})$, there are no edges from V_1 to V_2 and $|V_i| \leq 2n/3$ for $i \in \{1, 2\}$. The algorithm finds such a partition and guesses how D interacts with X.

In particular, first the algorithm correctly guesses $D' = D \cap X$ (by looping over all subsets of X). For each guess, it puts $N(D')$ into S and removes D' and $S \cap X$ from G (these vertices are already dominated and will not be used in the future to dominate even more vertices). For every remaining vertex v in X, the algorithm guesses whether it will be dominated by a vertex in V_1, in which case the algorithm deletes all edges from v to vertices in V_2, or by a vertex in V_2, in which case the algorithm deletes all edges from v to vertices in V_1. Let V_i' be V_i plus all the vertices in $X \setminus S$ that we guessed were dominated from V_i. At this point V_1' and V_2' are distinct components of the instance and can be solved independently. The running time is governed by the following recurrence.

$$T(n) = n^{O(1)} \cdot 2^{\mathcal{O}(\sqrt{n})} \cdot 2 \cdot T(2n/3) = 2^{\mathcal{O}(\sqrt{n})}$$

The space used is clearly polynomial. This concludes the proof. □

4 Kernel for CONNECTED DOMINATING SET

In this section we give a linear kernel for CONNECTED DOMINATING SET (CDS). Just as in the kernelization algorithm for DS, the kernelization for CDS is a recursive procedure. However the correctness proof of Irrelevant Vertex Rule is more complicated and requires more care. As for DS, we will apply this reduction rule in a divide and conquer manner to obtain a treewidth deletion set of size $\mathcal{O}(k)$. Then, by applying the "protrusion rule" together with the fact that CDS has finite integer index, we obtain the desired linear kernel for CDS.

Reducing the treewidth of a torso. As with DS, we are reducing the treewidth of a torso not only in the beginning of the procedure but also when we apply a recursive procedure to obtain $\mathcal{O}(k)$ sized treewidth deletion set. Let G be an H-minor free graph, S be a dominating set of G (not necessarily connected), L_t be one of its torsos, and A, $|A| \leq h$, be the set of apices of L_t, where h is some constant depending only on H. We will define a reduction rule that essentially "preserves" all dominating sets of size at most $3|S| + 3$ with "good enough" connectivity properties, without introducing new such sets. Just as for DS we will say that a subset A' of A is feasible if the factor 2-approximation for DS on H-minor free graphs concludes that there exists a set D of size at most $6|S| + 6$ which dominates all but S, such that $S \cap A = A'$. If such a set exists and A' is feasible we denote this set by $D(A')$.

Recall, that for DS we had the notion of a representative element for every subset $A' \subseteq A$. The representative vertex was crucially used in establishing Lemma 1, where we used it to simulate all the domination properties of the deleted vertex "w". We need a similar notion of representatives for CDS, however here the representatives will be vertex subsets rather than single vertices. With vertex subsets we would be able to simulate not only domination properties, but also the connectivity properties of an irrelevant vertex. More precisely, for every subset $A' \subseteq A$, we compute a minimum size vertex set $T \subseteq V(G) \setminus A$ such that $G[T]$ is connected and $A' \subseteq N[T]$. If the size of such a minimum set is at most $4h$, then we say that $T = T(A')$ is a *representative* of A', and add all the vertices in T to the set R. Note that $|R| \leq 4h \cdot 2^h$. For each A' we can test whether a representative exists in time $2^{|A'|}n^{\mathcal{O}(1)} = 2^h n^{\mathcal{O}(1)}$ by making a modification of the algorithm for the Steiner tree problem from [5]. Alternatively we can test it in time $n^{4h+O(1)}$ by brute force. The set of vertices *covered* by A' is $W(A') = N[A'] \setminus (A \cup S)$. Note that a vertex in S is never covered by a set A'. The definition of an irrelevant vertex with respect to A is different than for DS. A vertex w is called *irrelevant with respect to* A', if $N^{4h}_{G \setminus A}[w] \subseteq W(A')$. Here $N^{4h}_{G \setminus A}[w]$ is the set of vertices at distance at most $4h$ from w in the graph $G \setminus A$ (not in G). The irrelevant vertex rule for CDS is exactly the same as in Section 3 for DS but the corretness proof and analysis is more complicated.

Irrelevant Vertex Rule: If a vertex w is irrelevant with respect to every feasible $A' \subseteq A$ then delete w from G.

Just as for DS we will apply the irrelevant vertex rule in a recursive manner, and because of this it is not sufficient to prove that the rule is just solution preserving. In fact we need to prove that the rule preserves all sets that are "close enough" to being feasible solutions (and does not introduce new ones). We will again define what it means for a pair (G', S') to *simulate* (G, S). This definition is a bit lengthy and we split it in two parts.

Definition 3. *Let G be a graph, G' be an induced subgraph of G, S be a dominating set of G and $S \subseteq S' \subseteq V(G')$*

- *A set Z in G is called* interesting *if $|Z| \leq 3|S|$, Z dominates everything but S and every connected component of $G[Z]$ contains at least one vertex of S. Similarly, a set Z' in G' is called* interesting *if $|Z'| \leq 3|S'|$, Z' dominates everything but S', and every connected component of $G[Z']$ contains at least one vertex of S'.*
- *A set Z' in G' is called a* companion *to a set Z in G if $|Z'| \leq |Z|$, Z' dominates all the vertices of $N_G[Z] \cap V(G')$, $(Z \cap S) \subseteq Z'$, and for every connected component C' of $G'[Z']$ there exists a connected component C in $G[Z]$ such that $C \cap S \subseteq C' \cap S$.*
- *A set Z in G is called* companion *to a set Z' in G', if $|Z| \leq |Z'|$, it dominates all the vertices of $N[Z'] \cup (V(G) \setminus V(G'))$ in G, $(Z' \cap S) \subseteq Z$ and every connected component C of $G[Z]$ there exists a connected component C' in $G'[Z']$ such that $C' \cap S \subseteq C \cap S$.*

Definition 4. *Let G be a graph, G' be an induced subgraph of G, S be a dominating set of G and $S \subseteq S' \subseteq V(G)$ and $S \subseteq V(G) \cap V(G')$. We say that (G', S')* simulates (G, S) *if (i) for every interesting $Z \subseteq V(G)$, there is a companion $Z' \subseteq V(G')$, and (ii) for every interesting $Z' \subseteq V(G')$, there is a companion $Z \subseteq V(G)$.*

Just as for DS, it holds that for $S' \supseteq S$ we have that (G, S') simulates (G, S), and again we have transitivity of simulation. In particular it is easy to verify that if (G'', S'') simulates (G', S') and (G', S') simulates (G, S) then (G'', S'') simulates (G, S).

Let us also remark that if (G', S') simulates (G, S), then for every $p \leq 3|S|$, G has a connected dominating set Z of size at most p if and only if G' does. In particular consider a connected dominating set Z of G of size at most $p \leq 3|S|$. Set Z is interesting and so it has a companion Z' in G'. Set Z' clearly dominates G' and it remains to prove that Z' is connected. Suppose $G[Z]$ has two distinct components C_1 and C_2. However, $Z \cap S \subseteq C_1$ and $Z \cap S \subseteq C_2$, hence C_1 and C_2 have a non-empty intersection, which is a contradiction. If G' has a connected dominating set Z', then we can obtain Z from Z' in a similar fashion. Here we cheated a little bit, because we assumed that $Z \cap S \neq \emptyset$. This can be easily overcome by adding for some vertex v, its closed neighbourhood $N[v]$ to S. Every H-minor free graph is h-degenerated for some constant h depending only on H, and there is a vertex of degree h. Thus adding $N[v]$ to S does not affect the size of S much. We now prove correctness of the irrelevant vertex rule.

Lemma 9. *Let S be a dominating set in G, and $G' = G \setminus \{w\}$ be a graph obtained by applying the Irrelevant Vertex Rule on G. Then (G', S) simulates (G, S).*

Proof. We prove that every interesting Z in G has a companion in G'. Let $Z \subseteq V(G)$ be such that $|Z| \leq 3|S|$, Z dominates everything but S in G and every component of $G[Z]$ contains a vertex from S. Let $A' = Z \cap A$, and observe that A' is feasible since Z dominates all but S. If $w \notin Z$, then $Z' = Z$ is a companion of Z in G' and we are done. So assume $w \in Z$. Since w is irrelevant with respect to A' we have that $N^{4h}_{G \setminus A}[w] \subseteq W(A')$.

Let X be the vertex set of the connected component of $G[Z \cap N^{4h}_{G \setminus A}[w]]$ that contains w, and let C be the component of $G[Z]$ that contains w. If $|X| < 4h$ then there is a subset $X' = T(N(X) \cap A)$ such that $X' \subset R$, $|X'| \leq |X|$, $G[X']$ is connected and $N(X') \cap A \supseteq N(X) \cap A$. Consider $Z' = (Z \setminus X) \cup X'$ and let C' be the component of Z' that contains X'. Since $X \subseteq W(A')$, we have that $C \cap S \subseteq C' \cap S$ and $N(C) \subseteq N(C')$. Furthermore since $X' \subseteq R$, Z' avoids w and so Z' is a companion to Z in $G' = G \setminus \{w\}$.

Now suppose that $|X| \geq 4h$. Let $A^* = N(X) \cap A$. The vertex set A^* is a dominating set of size at most h in the graph $G[A^* \cup X]$ and so $G[A^* \cup X]$ has a connected dominating set X^* that contains A^* of size at most $3h$. Let P be the connected component of $G[X^*] \setminus A$ that contains w. Notice that $|P| \leq 2h$ and so there is a connected set $P' \subseteq R$ such that $|P'| \leq |P|$ and $N(P) \cap A \subseteq N(P') \cap A$. Finally, let Y be the set of vertices in X that are at distance exactly $4h$ from w in $G \setminus A$. Note that $X \setminus Y \geq 4h - 1$ and that $N[Y] \cap A \subseteq A^*$. Set $X' = X^* \setminus P \cup P'$, and $Z' = (Z \setminus (X \setminus Y)) \cup X'$. We have that $|X'| \leq |X^*| \leq 3h$ while $|X \setminus Y| \geq 4h - 1 \geq 3h$. Hence $Z' \leq Z$. Let C' be the component of Z' that contains X'. Since $X \subseteq W(A')$ it follows that $C \cap S \subseteq C' \cap S$ and $N(C) \subseteq N(C')$. Furthermore because $P' \subseteq R$, Z' avoids w and so Z' is a companion to Z in $G' = G \setminus \{w\}$.

For the second condition of simulation, let $Z' \subseteq V(G')$ such that $|Z'| \leq 3|S|$, Z' dominates everything but S in G' and each component of $G[Z']$ contains a vertex in S. We show that Z' also dominates w in G. Specifically $Z' \cup \{w\}$ is a dominating set of all but S in G of size at most $|S| + 1$ so $Z' \cap A$ is feasible. Since $\{w\}$ is irrelevant with respect to $Z' \cap A$ we have $w \in N_G(Z' \cap A)$. This concludes the proof. $\square$

Just as for DS, it is possible to prove that after removing all irrelevant vertices, the treewidth of each torso in the reduced graph is $\mathcal{O}(\sqrt{|S|})$. The most important difference is that instead of 2-dominating set we construct a $(4h + 1)$-dominating set in the proof.

Lemma 10. *Let G be a graph which is irreducible by the Irrelevant Vertex Rule and S be a dominating set of G. For every torso L_t of G, $\mathbf{tw}(L_t) = \mathcal{O}(\sqrt{|S|})$.*

Proof. Let $L_t^* = L_t \setminus A$, where A are the apices of L_t. We will obtain a $(4h + 1)$-dominating set of size $\mathcal{O}(|S|)$ in L_t^*. Towards this end, consider the following set, $Q = \bigcup_{A' \subseteq A, A' \text{is feasible}} D(A') \cup R \cup S \setminus A$. The size of the set of representative vertices, R, is at most $4h \cdot 2^{|A|} \leq 4h \cdot 2^h$. The number of feasible subsets A' is at most 2^h, where h is a constant depending only on H. The size of $D(A')$ is at most $6|S| + 6$ for every A'. Thus $|Q| \leq 2^h(6|S| + 6) + 4h \cdot 2^h + |S| = \mathcal{O}(|S|)$. We prove that Q is a $(4h + 1)$-dominating set of $V(G) \setminus A$. Let $w \in V(G) \setminus A$. If $w \in R$ or $w \in S$ then Q dominates S. So suppose $w \notin R \cup S$. Then, since w is not irrelevant there is a feasible subset A' of A such that w is relevant with respect to A'. Hence there exists a vertex w' in $N^4_{G \setminus A}h[w]$ which is not in $W(A')$. If $w' \in S$ then w is $4h$-dominated by $w' \in Q$ in $G \setminus A$. Otherwise w' is dominated by some w'' in $D(A')$ and hence w is $4h + 1$-dominated by $w'' \in Q$ in $G \setminus A$. Hence $G \setminus A$ has a $(4h + 1)$-dominating set of size $\mathcal{O}(|S|)$.

The graph L_t^* can be obtained from $G \setminus A$ by contracting all edges in $E(G \setminus A) \setminus E(L_t^*)$ and adding all edges in $E(L_t^*) \setminus E(G \setminus A)$. Since contracting and adding edges can not increase the size of a minimum $(4h + 1)$-dominating set of a graph, L_t^* has a $(4h + 1)$-dominating set of size $\mathcal{O}(|S|)$.

To conclude, L_t^* excludes an apex graph as a minor (see discussions after Theorem 1) and it has a $(4h + 1)$-dominating set of size $\mathcal{O}(|S|)$. By the bidimensionality of $(4h + 1)$-dominating set, we have that $\mathbf{tw}(L_t^*) = \mathcal{O}(\sqrt{|S|})$ [16, 27]. Now we add all the apices of A to all the bags of the tree decomposition of L_t^* to obtain a tree decomposition for L_t'. Thus $\mathbf{tw}(L_t') \leq \mathcal{O}(\sqrt{|S|}) + h = \mathcal{O}(\sqrt{|S|})$. $\square$

Applying the irrelevant vertex rule exhaustively on all torsos, and bounding treewidth using Lemma 10, we arrive at the following lemma.

Lemma 11. *There is a polynomial time algorithm that for a given graph G and a dominating set S of G, outputs an induced subgraph G' of G such that(G', S) simulates (G, S) and $\mathbf{tw}(G') = \mathcal{O}(\sqrt{|S|})$.*

We proceed as follows. Given a *connected* graph G and a positive integer k, we first apply factor 2-approximation algorithm given in [17, 28] for DS on G and obtain a dominating set S. If the size of S is more than $2k$ then we return that G does not have a connected dominating set of size at most k. If the size of S is at most $2k$, we proceed further. To prove Lemma 13, we need an additional property of S, namely that every dominating set *contains at least one vertex from S*. To ensure that S has this property, we choose a vertex v of minimum degree and add $N[v]$ to S. Since G excludes a fixed graph H as a minor there exists a constant c such that G is $p = c|V(H)|\sqrt{\log |V(H)|}$ degenerate [20]. This implies that the degree of v in G is at most p and hence $|S| \leq 2k + p + 1 = \mathcal{O}(k)$.

The recursive procedure to obtain a treewidth deletion set of size $\mathcal{O}(k)$ is almost identical to the one for DS. The main difference is that it is slightly more complicated to ensure proper simulation when splitting the graph into two independent subproblems.

Lemma 12. *Let G be an H-minor free graph and S be a dominating set of G. There is a polynomial time algorithm that computes an induced subgraph G^* of G, a dominating set S^* of G^* and a set D^* such that (G^*, S^*) simulates (G, S) and $\mathbf{tw}(G^* \setminus D^*) = \mathcal{O}(1)$ and $|D| = \mathcal{O}(|S|)$.*

Proof. By Lemma 11 we may assume that $\mathbf{tw}(G) = \mathcal{O}(\sqrt{|S|})$. Hence, if $|S| = \mathcal{O}(1)$ we are done, as we can return (G, S) with $D^* = \emptyset$. Otherwise, using a constant factor approximation of treewidth on H-minor-free graphs [25], we compute a tree-decomposition of G of width $d\sqrt{k}$. Now, by applying Lemma 4 on this decomposition we find a partitioning of $V(G)$ into V_1, V_2 and X such that there are no edges from V_1 to V_2, $|X| \le d\sqrt{k} + 1$ and $|V_i \cap S| \le 2|S|/3$ for $i \in \{1, 2\}$. Let $S' = S \cup X$, and observe that (G, S') simulates (G, S).

We now apply the algorithm recursively on $(G[V_1 \cup X], S' \cap (V_1 \cup X))$ and $(G[V_2 \cup X], S' \cap (V_2 \cup X))$ and obtain graphs G_1, G_2 and sets S_1, S_2, D_1, D_2 such that for $i \in \{1, 2\}$

- (G_i, S_i) simulates $(G[V_i \cup X], S' \cap (V_i \cup X))$
- $\mathbf{tw}(G_i \setminus D_i) = \mathcal{O}(1)$.

Observe that since $X \subseteq S'$ we have that $S' \cap (V_i \cup X)$ is indeed a dominating set of $G[V_i \cup X]$ and hence we actually can run the algorithm recursively on the two subcases. The algorithm will return $G^* = G[V(G_1) \cup V(G_2)]$ and the sets $S^* = S_1 \cup S_2$ and $D^* = D_1 \cup D_2 \cup X$. Clearly $\mathbf{tw}(G^* \setminus D^*) = \mathcal{O}(1)$ so we need to prove that (G^*, S^*) simulates (G, S') and that $|D^*| = \mathcal{O}(|S|)$.

Claim 2. *(G^*, S^*) simulates (G, S')*

Proof. It follows directly from the construction that $S' \subseteq S^*$ so it suffices to prove conditions (i) and (ii) of simulation. For (i), let Z dominate all but S' in G, and $|Z| \le 3|S'|$. For $i \in \{1, 2\}$ set $Z_i = Z \cap (V_i \cup X)$. We also assume that $|Z_i| \le 3|S' \cap (V_i \cup X)|$. This need not be true in general, and in the case when $|Z_i| > 3|S' \cap (V_i \cup X)|$, we will take $Z_i = S' \cap (V_i \cup X)$. Now we prove for the case when it holds than $|Z_i| \le 3|S' \cap (V_i \cup X)|$ for $i \in \{1, 2\}$, other cases are analogous.

Since (G_i, S_i) simulates $(G[V_i \cup X], S' \cap (V_i \cup X))$ it follows that there are sets Z_1' and Z_2' such that for $i \in \{1, 2\}$, Z_i' dominates $N_G[Z_i] \cap V(G_i)$ in G_i, $|Z_i'| \le |Z_i|$ and $Z_i \cap S' \subseteq Z_i'$. Set $Z' = Z_1' \cup Z_2'$. We have that

$$(Z \cap S) = (Z_1 \cap S) \cup (Z_2 \cap S) \subseteq (Z_1 \cap S_1) \cup (Z_2 \cap S_2) \subseteq Z_1' \cup Z_2' \subseteq Z'.$$

Furthermore, Z' dominates $N_G[Z] \cap (V(G_1) \cup V(G_2)) = N_G[Z] \cap V(G^*)$. The set Z' dominates $N_G[Z] \cap V(G^*)$ in G^* because $N[Z] = N[Z_1] \cup N[Z_2]$ and Z_1' dominates $N[Z_1]$ and Z_2' dominates $N[Z_2]$. We have that $|Z'| \le |Z|$ because

$$\begin{aligned}
|Z'| = |Z_1' \cup Z_2'| &= |Z_1'| + |Z_2'| - |Z_1' \cap Z_2'| \\
&\le |Z_1| + |Z_2| - |Z_1' \cap Z_2'| \\
&= |Z_1 \cup Z_2| + |Z_1 \cap Z_2| - |Z_1' \cap Z_2'| \\
&= |Z_1 \cup Z_2| + |Z \cap X| - |Z_1' \cap Z_2'| \\
&\le |Z_1 \cup Z_2| = |Z|.
\end{aligned}$$

Finally, we show that for every connected component C' of $G^*[Z']$ there exists a connected component C in $G[Z]$ such that $C \cap S' \subseteq C' \cap S'$. Let $C_1^Z, \ldots, C_\ell^Z$ be the connected components of $G[Z]$. Now given a connected component C_i we call it a *broken component* if $C_i^Z \cap X \neq \emptyset$. In other words, these are components of $G[Z]$ intersecting both Z_1 and Z_2. Else, we call a component *non-broken*. Now let C' be a connected component of $G^*[Z']$. Let $\mathcal{F}(C') = \{E_1, E_2, \ldots, E_\ell\}$ be the connected components of C' in $G_1[Z_1']$ and $G_2[Z_2']$. That is, we look at the restriction of C' in $G_1[Z_1']$ and $G_2[Z_2']$ and $\mathcal{F}(C')$ is the set of connected components in any of them. Thus, each of E_i is a connected component of one of $G_j[Z_j']$, $j \in \{1, 2\}$. Let us fix a connected component E_i, say it is in $G_1[Z_1']$. Now, by induction hypothesis we know that there exists a connected component F_i in $G[Z_1]$ such that $F_i \cap (S' \cap (V_1 \cup X)) \subseteq E_i \cap (S' \cap (V_1 \cup X))$. If for any E_i we have F_i such that F_i is a non-broken component of $G[Z]$ then we associate F_i to C'. Clearly, we have that $F_i \cap (S' \cap (V_1 \cup X)) = F_i \cap S' \subseteq C' \cap S'$. Thus, we assume that each of F_i is a broken component.

For a given a connected component C_j^Z, we define the set $\mathcal{F}(C_j^Z)$ as the set of all the connected components we get out of C_j^Z when we restrict either to $G[Z_1]$ or $G[Z_2]$. We call the connected components in $\mathcal{F}(C_j^Z)$ as *broken pieces*. Observe that every broken piece contains a vertex of X. In fact, for every vertex $v \in X$, such that there is some component A_L in $G[Z_1]$ containing v, there is another component B_L in $G[Z_2]$ containing v. Furthermore, $v \in X$ is in exactly two connected components in $\mathcal{F}(C_j^Z)$. Observe that we could make similar remarks about the components in $\mathcal{F}(C')$. Combing back to the proof, we have that each of F_i is a broken piece. This implies that for all i, we have that $(F_i \cap X) \neq \emptyset$ and since $(F_i \cap X) \subseteq S'$ we have that $(F_i \cap X) \subseteq E_i$. Let C_j^Z be the connected component whose one of the broken piece is F_1. Now with C' we associate C_j^Z. Now the only thing remaining to show is that $C_j^Z \cap S' \subseteq C' \cap S'$.

Consider the following auxiliary bipartite graph $\mathcal{B}$ with parts M and $N = C_j^Z \cap X$. In M we have a vertex for every connected component in $\mathcal{F}(C_j^Z)$. We gave an edge between a component vertex and a vertex in N if that vertex is part of the corresponding component. Clearly, $\mathcal{B}$ is connected, since C_j^Z is a connected component. Also notice that every vertex in N has degree 2. Let T be a breadth first search (BFS) tree of $\mathcal{B}$ rooted at the vertex corresponding to F_1. Recall that we have E_1 associated with F_1. Notice that every leaf of this tree is a vertex corresponding to a connected component and every alternate layer starting from the root has only degree 2 vertices. We will show using a traversal of T, that for every connected component P in $\mathcal{F}(C_j^Z)$ there exists a component Q in $\mathcal{F}(C')$ such that $(P \cap S') \subseteq Q \cap S'$. This in turn implies that $C_j^Z \cap S' \subseteq C' \cap S'$.

We traverse the tree T top down and to each of the vertices in T, we will associate a component from $\mathcal{F}(C')$. The root has been associated with E_1. Consider the root and its children, by our construction we know that these vertices are in both F_1 and E_1. Consider the children of the root. These are degree two vertices. Recall, that every vertex is part of exactly 2 components in $\mathcal{F}(C')$, thus we assign to this degree two vertex a component that has not been assigned before. Now if we have a vertex in tree corresponding to a component vertex then we just associate with it the component of its parent, which happens to be a degree 2 vertex. We recursively assign components to each of the vertices in the tree. This gives an unique assignment starting from the root. Now we only need to show that if a component F_j (some non-root component) has been assigned some E_p then $(F_j \cap S') \subseteq E_p \cap S'$. Let $u \in X$ be the parent of F_j in the tree. Notice that by our construction we have that if F_j is contained in $G[V_i \cup X]$ then E_p is contained in G_i. However, by induction hypothesis we know that there is some connected component C^* in $G[Z_i]$ such that $C^* \cap S' \subseteq E_p \cap S'$. Notice that in $G[Z_i]$ there is a unique component that contains u and we also know that $Z_i \cap S' \subseteq Z_i'$, thus this implies that the component C^* has to be F_j. This completes the proof in one direction.

By arguments identical to those in the forward direction, we can prove that condition (ii) of simulation holds. This concludes the proof of the claim. ☐

Finally, we upper bound $|D^*|$ by the following recursive formula.

$$|D^*| \leq \max_{1/3 \leq \alpha \leq 2/3} \left\{ \mu\left(\alpha|S| + d\sqrt{|S|}\right) + \mu\left((1-\alpha)|S| + d\sqrt{|S|}\right) + d\sqrt{|S|} \right\}.$$

Using simple induction one can show that the above solves to $\mathcal{O}(|S|)$. See for an example [28, Lemma 2]. Hence we conclude that $|D| = \mathcal{O}(|S|) = \mathcal{O}(k)$. This completes the proof of the lemma. ☐

Lemma 12 establishes that the output graph G^* and set S^* simulates (G, S). Since simulation implies that for every $p \leq 3|S| + 3$, G^* has a dominating set of size p if and only if G does, the set S was obtained by running a 2-approximation for dominating on G, and the size of a minimum connected dominating set in a connected graph is at most thrice the size of the minimum dominating set, we get the following lemma.

Lemma 13. *Let G be an H-minor free graph and S be a dominating set of G. There is a polynomial time algorithm that computes an induced subgraph G^* and a set D^* of size $\mathcal{O}(|k|)$ such that $\mathbf{tw}(G^* \setminus D^*) = \mathcal{O}(1)$ and G^* has a connected dominating set of size k if and only if G does.*

Finally, CDS has finite integer index [8] and the statement similar to Lemma 8 for CDS is a special case of [29, Lemma 4.1]. Now using Lemmata 7 and 13, we can show the following theorem along the lines of Theorem 3.

Theorem 5. *Let $\mathcal{G}_H$ be the class of graphs excluding a fixed graph H as a minor then* CONNECTED DOMINATING SET *has linear kernel on $\mathcal{G}_H$.*

Let us observe, that Theorem 5 combined with the standard dynamic programing on graphs of bounded treewidth implies that CDS on H-minor free graphs is solvable in time $2^{\mathcal{O}(\sqrt{k}\log k)} + n^{\mathcal{O}(1)}$. Recently, Cygan et al. [10] gave a Monte Carlo algorithm solving CDS on graphs of treewidth t in time $3^t n^{\mathcal{O}(1)}$. Combined with linear kernel, it results in the running time $2^{\mathcal{O}(\sqrt{k})} + n^{\mathcal{O}(1)}$. To our knowledge, these are the first subexponential parameterized algorithm for CDS on H-minor free graphs.

5 Conclusions

We conclude with several open questions. It is tempting to ask if the kernelization framework on apex-minor free graphs developed in [29] for contraction bidimensional problems with separation properties can be extended to minor free graphs. This question remains open even for r-domination with $r > 1$. Another natural question is if the linear kernel for DS can be obtained for more general classes. H-minor free graphs form a general class of sparse graph but DS is known to be FPT even on more general classes of sparse graphs like graphs locally excluding some graph as a minor, degenerated graphs, graphs of bounded expansions, and nowhere dense classes of graphs [12, 13, 24, 39]. A word of caution is appropriate here: there are classes of sparse graphs where existence of a linear kernel for DS is highly unexpected. For example, an easy reduction from the result of Dell and van Melkebeek from [14] that d-HITTING SET has no kernel of size $k^{d-\varepsilon}$ for any $\varepsilon > 0$ unless coNP is in NP/poly, shows that DS has no kernel of size $k^{d-\varepsilon}$ on d-degenerate graphs. For CDS the situation is even worse, by the recent result of Cygan et al. [11], the problem does not have a polynomial kernel on d-degenerated graphs unless coNP is in NP/poly.

References

[1] J. ALBER, H. L. BODLAENDER, H. FERNAU, T. KLOKS, AND R. NIEDERMEIER, *Fixed parameter algorithms for dominating set and related problems on planar graphs*, Algorithmica, 33 (2002), pp. 461–493.

[2] J. ALBER, M. R. FELLOWS, AND R. NIEDERMEIER, *Polynomial-time data reduction for dominating sets*, J. ACM, 51 (2004), pp. 363–384.

[3] N. ALON AND S. GUTNER, *Kernels for the dominating set problem on graphs with an excluded minor*, Tech. Rep. TR08-066, ECCC, 2008.

[4] N. ALON, P. SEYMOUR, AND R. THOMAS, *A separator theorem for nonplanar graphs*, J. Amer. Math. Soc., 3 (1990), pp. 801–808.

[5] A. BJÖRKLUND, T. HUSFELDT, P. KASKI, AND M. KOIVISTO, *Fourier meets Möbious: Fast subset convolution*, in Proceedings of the 39th annual ACM Symposium on Theory of Computing (STOC 2007), New York, 2007, ACM Press, pp. 67–74.

[6] H. L. BODLAENDER, *A partial k-arboretum of graphs with bounded treewidth*, Theoret. Comput. Sci., 209 (1998), pp. 1–45.

[7] H. L. BODLAENDER, R. G. DOWNEY, M. R. FELLOWS, AND D. HERMELIN, *On problems without polynomial kernels*, J. Comput. Syst. Sci., 75 (2009), pp. 423–434.

[8] H. L. BODLAENDER, F. V. FOMIN, D. LOKSHTANOV, E. PENNINKX, S. SAURABH, AND D. M. THILIKOS, *(Meta) Kernelization*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009), IEEE, 2009, pp. 629–638.

[9] J. CHEN, H. FERNAU, I. A. KANJ, AND G. XIA, *Parametric duality and kernelization: Lower bounds and upper bounds on kernel size*, SIAM J. Comput., 37 (2007), pp. 1077–1106.

[10] M. CYGAN, J. NEDERLOF, M. PILIPCZUK, M. PILIPCZUK, J. M. M. VAN ROOIJ, AND J. O. WOJTASZCZYK, *Solving connectivity problems parameterized by treewidth in single exponential time*, CoRR, abs/1103.0534 (2011).

[11] M. CYGAN, M. PILIPCZUK, M. PILIPCZUK, AND J. WOJTASZCZYK, *Kernelization hardness of connectivity problems in d-degenerate graphs*, in Proceedings of the 36th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2010), vol. 6410 of Lect. Notes Comp. Sc., Springer, 2010, pp. 147–158.

[12] A. DAWAR, M. GROHE, AND S. KREUTZER, *Locally excluding a minor*, in Proceedings of the 22nd IEEE Symposium on Logic in Computer Science (LICS 2007), IEEE Computer Society, 2007, pp. 270–279.

[13] A. DAWAR AND S. KREUTZER, *Domination problems in nowhere-dense classes*, in Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009), vol. 4 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2009, pp. 157–168.

[14] H. DELL AND D. VAN MELKEBEEK, *Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses*, in Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010), 2010, pp. 251–260.

[15] E. D. DEMAINE, F. V. FOMIN, M. HAJIAGHAYI, AND D. M. THILIKOS, *Fixed-parameter algorithms for (k, r)-center in planar graphs and map graphs*, ACM Trans. Algorithms, 1 (2005), pp. 33–47.

[16] ——, *Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs*, J. ACM, 52 (2005), pp. 866–893.

[17] E. D. DEMAINE AND M. HAJIAGHAYI, *Bidimensionality: new connections between FPT algorithms and PTASs*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), New York, 2005, ACM-SIAM, pp. 590–601.

[18] ——, *The bidimensionality theory and its algorithmic applications*, The Computer Journal, 51 (2007), pp. 332–337.

[19] E. D. DEMAINE, M. T. HAJIAGHAYI, AND K.-I. KAWARABAYASHI, *Approximation algorithms via structural results for apex-minor-free graphs*, in Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP 2009), vol. 5555 of Lect. Notes Comp. Sc., Springer, 2009, pp. 316–327.

[20] R. DIESTEL, *Graph theory*, vol. 173 of Graduate Texts in Mathematics, Springer-Verlag, Berlin, third ed., 2005.

[21] F. DORN, F. V. FOMIN, D. LOKSHTANOV, V. RAMAN, AND S. SAURABH, *Beyond bidimensionality: Parameterized subexponential algorithms on directed graphs*, in Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS 2010), vol. 5 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010, pp. 251–262.

[22] R. G. DOWNEY AND M. R. FELLOWS, *Parameterized Complexity*, Springer, 1998.

[23] F. F. DRAGAN, F. V. FOMIN, AND P. A. GOLOVACH, *Spanners in sparse graphs*, in Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008), vol. 5125 of Lect. Notes Comp. Sc., Springer, 2008, pp. 597–608.

[24] Z. DVORAK, D. KRÁL, AND R. THOMAS, *Deciding first-order properties for sparse graphs*, in Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010), IEEE Computer Society, 2010, pp. 133–142.

[25] U. FEIGE, M. HAJIAGHAYI, AND J. R. LEE, *Improved approximation algorithms for minimum weight vertex separators*, SIAM J. Comput., 38 (2008), pp. 629–657.

[26] J. FLUM AND M. GROHE, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.

[27] F. V. FOMIN, P. A. GOLOVACH, AND D. M. THILIKOS, *Contraction obstructions for treewidth*, J. Comb. Theory, Ser. B, 101 (2011), pp. 302–314.

[28] F. V. FOMIN, D. LOKSHTANOV, V. RAMAN, AND S. SAURABH, *Bidimensionality and EPTAS*, in Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011), SIAM, 2010, pp. 748–759.

[29] F. V. FOMIN, D. LOKSHTANOV, S. SAURABH, AND D. M. THILIKOS, *Bidimensionality and kernels*, in Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010), ACM-SIAM, 2010, pp. 503–510.

[30] F. V. FOMIN AND D. M. THILIKOS, *Dominating sets in planar graphs: Branch-width and exponential speed-up*, SIAM J. Comput., 36 (2006), pp. 281–309.

[31] M. GROHE, *Local tree-width, excluded minors, and approximation algorithms*, Combinatorica, 23 (2003), pp. 613–632.

[32] S. GUTNER, *Polynomial kernels and faster algorithms for the dominating set problem on graphs with an excluded minor*, in Proceedings of the 4th Workshop on Parameterized and Exact Computation (IWPEC 2009), Lect. Notes Comp. Sc., Springer, 2009, pp. 246–257.

[33] T. W. HAYNES, S. T. HEDETNIEMI, AND P. J. SLATER, *Fundamentals of domination in graphs*, Marcel Dekker Inc., New York, 1998.

[34] I.ADLER, S.G.KOLLIOPOULOS, P.K.KRAUSE, D.LOKSHTANOV, S.SAURABH, AND D.THILIKOS, *Tight bounds for linkages in planar graphs*, in Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP 2011), Lect. Notes Comp. Sc., Springer, 2011, p. to appear.

[35] K. ICHI KAWARABAYASHI AND P. WOLLAN, *A simpler algorithm and shorter proof for the graph minor decomposition*, in Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC 2011), ACM, 2011, pp. 451–458.

[36] K.-I. KAWARABAYASHI AND Y. KOBAYASHI, *The induced disjoint path problem*, in Proceedings of the 13th Conference on Integer Programming and Combinatorial Optimization (IPCO 2008), vol. 5035 of Lect. Notes Comp. Sc., Springer, Berlin, 2008, pp. 47–61.

[37] K.-I. KAWARABAYASHI AND B. REED, *Odd cycle packing*, in Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010), New York, NY, USA, 2010, ACM, pp. 695–704.

[38] Y. KOBAYASHI AND K.-I. KAWARABAYASHI, *Algorithms for finding an induced cycle in planar graphs and bounded genus graphs*, in Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2009), ACM-SIAM, 2009, pp. 1146–1155.

[39] J. NEŠETŘIL AND P. O. DE MENDEZ, *Grad and classes with bounded expansion II. Algorithmic aspects*, Eur. J. Comb., 29 (2008), pp. 777–791.

[40] R. NIEDERMEIER, *Invitation to fixed-parameter algorithms*, vol. 31 of Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, Oxford, 2006.

[41] G. PHILIP, V. RAMAN, AND S. SIKDAR, *Solving dominating set in larger classes of graphs: FPT algorithms and polynomial kernels*, in Proceedings of the 17th Annual European Symposium on Algorithms (ESA 2009), vol. 5757 of Lect. Notes Comp. Sc., Springer, 2009, pp. 694–705.

[42] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. XIII. The disjoint paths problem*, J. Comb. Theory Series B, 63 (1995), pp. 65–110.

[43] ——, *Graph minors. XVI. Excluding a non-planar graph*, J. Combin. Theory Ser. B, 89 (2003), pp. 43–76.

[44] J. M. M. VAN ROOIJ, H. L. BODLAENDER, AND P. ROSSMANITH, *Dynamic programming on tree decompositions using generalised fast subset convolution*, in Algorithms—ESA 2009, vol. 5757 of Lecture Notes in Comput. Sci., Springer, Berlin, 2009, pp. 566–577.