Linear kernels for (connected) dominating set on graphs with excluded topological subgraphs

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 topological minor. In other words, we give polynomial time algorithms that, for a given H-topological-minor free graph G and a positive integer k, output an H-topological-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.

Our results extend the known classes of graphs on which Dominating Set and Connected Dominating Set problems admit linear kernels. Prior to our work, it was known that these problems admit linear kernels on graphs excluding a fixed graph H as a minor. Moreover, for Dominating Set, a kernel of size $k^{c(H)}$, where $c(H)$ is a constant depending on the size of H, follows from a more general result on the kernelization of Dominating Set on graphs of bounded degeneracy. For Connected Dominating Set no polynomial kernel on H-topological-minor free graphs was known prior to our work. On the negative side, it is known that Connected Dominating Set on 2-degenerated graphs does not admit a polynomial kernel unless $\mathsf{coNP} \subseteq \mathsf{NP/poly}$.

Our kernelization algorithm is based on a non-trivial combination of the following ingredients

- The structural theorem of Grohe and Marx [STOC 2012] for graphs excluding a fixed graph H as a topological subgraph;
- A novel notion of protrusions, different that the one defined in [FOCS 2009];
- Reinterpretations of reduction techniques developed for kernelization algorithms for Dominating Set and Connected Dominating Set from [SODA 2012].

A protrusion is a subgraph of constant treewidth separated from the remaining vertices by a constant number of vertices. Roughly speaking, in the new notion of protrusion instead of demanding the subgraph of being of constant treewidth, we ask it to contain a *constant* number of vertices from a solution. We believe that the new notion of protrusion will be useful in many other algorithmic settings.

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

1 Introduction

Kernelization is an emerging technique in parameterized complexity. 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

[*]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).

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 computational approach both from practical and theoretical perspectives. 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. There are also interesting links between lower bounds on the sizes of kernels and classical computational complexity [8, 16, 24].

The Dominating Set (DS) problem together with its numerous variants, is one of the most classic and well-studied problems in algorithms and combinatorics [41]. 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 additionally demand the subgraph induced by the dominating set to be connected. A considerable part of the algorithmic study on these NP-complete problems has been focused on the design of parameterized and kernelization algorithms. In general, DS is $\mathsf{W}[2]$-complete and therefore it cannot be solved by a parameterized algorithm, unless an unexpected collapse occurs in the Parameterized Complexity hierarchy (see [23, 28, 45]) and thus also does not admit a kernel. However, there are interesting graph classes where *fixed-parameter tractable* (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, 17, 36] resulted in the creation of bidimensionality theory characterizing a broad range of graph problems that admit efficient approximate schemes, fixed-parameter algorithms or kernels on a broad range of graphs [18, 20, 22, 31, 33, 32].

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 of Alber et al. [2], see also [12], has been extended to a much more general graph classes like graphs of bounded genus [9] and apex-minor free graphs [33]. An important step in this direction was done by Alon and Gutner [3, 40] who obtained a kernel of size $O(k^h)$ for DS on H-minor-free and H-topological-minor free graphs, where the constant h depends on the excluded graph H. Later, Philip, Raman, and Sikdar [46] 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. In particular, for d-degenerate graphs, a subclass of $K_{i,j}$-free graphs, the algorithm of Philip, Raman, and Sikdar [46] produces a kernel of size $\mathcal{O}(k^{d^2})$. Similarly, the sizes of kernels in [3, 40, 46] are bounded by polynomials in k with degrees depending on the size of the excluded minor H. Alon and Gutner [3] mentioned as a challenging question to characterize the families of graphs for which the dominating set problem admits a linear kernel, i.e. a kernel of size $f(h) \cdot k$, where the function f depends *exclusively* on the graph family. In this direction, there are already results for more restricted graph classes. According to the meta-algorithmic results on kernels introduced in [9], DS has a kernel of size $f(g) \cdot k$ on graphs of genus g. An alternative meta-algorithmic framework, based on bidimensionality theory [18], was introduced in [33], implying the existence of a kernel of size $f(H) \cdot k$ for DS on graphs excluding an apex graph H as a minor. Recently, the result on linear kernels on apex-minor-free graphs was extended to graphs excluding an arbitrary graph H as a minor [34]. Prior to our work, the only result on linear kernels for DS on graphs excluding H as a topological subgraph, was the result of Alon and Gutner in [3] for a very special case

$H = K_{3,h}$. See Fig. 1 for the relationship between these classes.

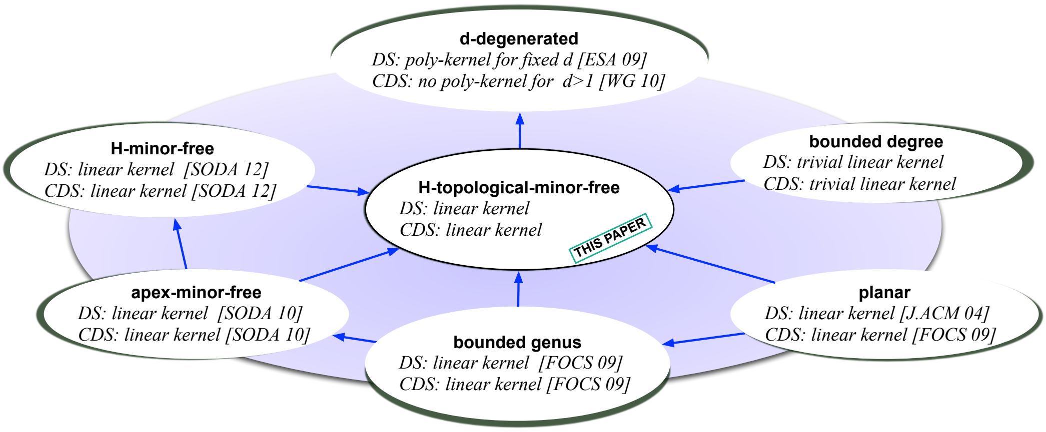

Figure 1: Kernels for DS and CDS on classes of sparse graphs. Arrows represent inclusions of classes. In the diagram, [J.ACM 04] is referred to the paper of Albers et al. [2], [FOCS 09] to the paper of Bodlaender et al. [9], [SODA 10] and [SODA 12] to the papers of Fomin et al. [33] and [34], [ESA 09] to the paper of Philip et al. [46], and [WG 10] to Cygan et al. [14].

It is tempting to suggest that similar improvements on kernel sizes are possible for more general graph classes like d-degenerated graphs. For example, for graphs of bounded vertex degree, a subclass of d-degenerate graphs, DS has a trivial linear kernel. Unfortunately, for d-degenerate graphs the existence of a linear kernel and even polynomial kernel with the exponent of the polynomial independent of d is very unlikely. By the very recent work of Cygan et al. [13], the kernelization algorithm of Philip, Raman, and Sikdar [46] is essentially tight—existence of a kernel of size $\mathcal{O}(k^{(d-3)(d-1)-\varepsilon})$, would imply that coNP is in NP/poly. In spite of these negative news, we show how to lift the linearity of kernelization for DS from bounded-degree graphs and H-minor free graphs to the class of graphs excluding H as a topological subgraph. Moreover, a modification of the ideas for DS kernelization can be used to obtain a linear kernel for CDS, which is usually a much more difficult problem to handle due to the connectivity constraint. For example, CDS does not have a polynomial kernel on 2-degenerated graphs unless coNP is in NP/poly [14].

The class of graphs excluding H as a topological subgraph is a wide class of graphs containing H-minor-free graphs and graphs of constant vertex degrees. The existence of a linear kernel for DS on this class of graphs significantly extends and improves previous works [3, 34, 40]. The basic idea behind kernelization algorithms on apex-minor-free and minor-free graphs is the bidimensionality of DS. Roughly speaking, the treewidth of these graphs with dominating set k is either $o(k)$ (as in planar, bounded genus or apex-minor-free graphs [18]) or becomes $o(k)$ after applying the irrelevant vertex technique [34]. This idea can hardly work on graphs of bounded degree, and hence on graphs excluding H as a topological subgraph. The reason is that the bound $o(k)$ on the treewidth of such graphs would imply that DS is solvable in subexponential time on graphs of bounded degree, which in turn can be shown to contradict the Exponential Time Hypothesis [42]. This is why the kernelization techniques developed for H-minor-free graphs does not seem to be applicable directly in our case.

High level overview of the main ideas. Our kernelization algorithm has two main phases. In the first phase we partition the input graph G into subgraphs $C_0, C_1, \ldots, C_\ell$, such that $|C_0| = \mathcal{O}(k)$; for every $i \geq 1$, the neighbourhood $N(C_i) \subseteq C_0$, and $\sum_{1\leq i\leq \ell} |N(C_i)| = \mathcal{O}(k)$. In the second phase, we replace these graphs by smaller equivalent graphs. Towards this, we treat graphs $N[C_i] = C_i \cup N(C_i)$, $i \geq 1$, as t-boundaried graphs with boundary $N(C_i)$. Our first conceptual contribution is a polynomial time algorithmic procedure replacing a t-boundaried graph by an equivalent graph of size $\mathcal{O}(|N(C_i)|)$. Observe that as a result of such replacements, the size of the new graph is $\sum_{1 \leq i \leq \ell} |\mathcal{O}(N(C_i))| + |C_0| = \mathcal{O}(k)$ and thus we obtain a linear kernel. Kernelization techniques based on replacing a t-boundaried graph by an equivalent instance, or more specifically, protrusion replacement, were used before [9, 33, 30, 44]. At this point it is also important to mention earlier works done in [27, 5, 7, 11, 15, 10] on protrusion replacement in the algorithmic setting on graphs of bounded treewidth. The substantial differences with our replacement procedure and the ones used before in the kernelization setting are the following.

- In the protrusion replacement procedure it is assumed that the size of the boundary t and the treewidth of the replaced graph are constants. In our case neither the treewidth, nor the boundary size are bounded. In particular, the boundary size could be a linear function of k.

- In earlier protrusion replacements, the size of the equivalent replacing graph is bounded by some (non-elementary) function of t. In our case this is a linear function of t.

Our new replacement procedure strongly exploits the fact that graphs C_i possess a set of desired properties allowing us to apply the irrelevant vertex technique from [34]. However, not every graph G excluding some fixed graph as a topological minor can be partitioned into graphs with the desired properties. We show that in this case there is another polynomial time procedure transforming G into an equivalent graph, which in turn can be partitioned. The procedure is based on a generalised notion of protrusion, which is the second conceptual contribution of this paper. In the new notion of protrusion we relax the requirement that protrusion is of bounded treewidth by the condition that it has a bounded dominating set. Let us remark, that a similar notion of a generalised protrusion bounded by the size of a certificate, can be used for a variety of graph problems. We show that either a graph does not have the desired partition, or it contains a sufficiently large generalised protrusion, which can be replaced by a smaller equivalent subgraph. For constructing the partitioning, we also devise a constant factor approximation algorithm for DS on graphs excluding some fixed graph as a topological minor. The construction of the partitioning as well as the approximation algorithm, are heavily based on the recent work of Grohe and Marx on the structure of such graphs [39].

2 Preliminaries

In this section we give various definitions which we make use of in the paper. We refer to Diestel's book [21] 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 the *subgraph induced by* V' if $E(G') = \{uv \in E(G) \mid u, v \in V'\}$. By $N_G(u)$ we denote the (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. Also for given graph G and a vertex subset S by $K[S]$ we mean a clique on the vertex set S. 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 $N^r_G(P)$ 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.

A *subdivision* of a graph H is obtained by replacing each edge of H by a non-trivial path. We say that H is a *topological minor* of G if some subgraph of G is isomorphic to a subdivision of H and denote it by $H \preceq_T G$. A graph G *excludes graph H as a (topological) minor* if H is not a (topological) minor of G. For a graph H, by $\mathcal{C}_H$, we denote all graphs that exclude H as topological minors.

Tree Decompositions. A *tree decomposition* of a graph $G = (V, E)$ is a pair (M, Ψ) where M is a rooted tree and $\Psi : V(M) \to 2^V$, such that :

1. $\bigcup_{t \in V(M)} \Psi(t) = V$.
2. For each edge $(u, v) \in E$, there is a $t \in V(M)$ such that both u and v belong to $\Psi(t)$.
3. For each $v \in V$, the nodes in the set $\{t \in V(M) \mid v \in \Psi(t)\}$ form a subtree of M.

If M is a path then we call the pair (M, Ψ) as *path decomposition*.
The following notations are the same as that in [39]. Given a tree decomposition of graph $G = (V, E)$, we define mappings $\sigma, \gamma : V(M) \to 2^V$ and $\kappa : E(M) \to 2^V$. For all $t \in V(M)$,

$$\sigma(t) = \begin{cases} \emptyset & \text{if } t \text{ is the root of } M \\ \Psi(t) \cap \Psi(s) & \text{if } s \text{ is the parent of } t \text{ in } M \end{cases}$$

$$\gamma(t) = \bigcup_{u \text{ is a descendant of } t} \Psi(u)$$

For all $e = uv \in E(M)$, $\qquad \kappa(e) = \Psi(u) \cap \Psi(v)$.
For a subgraph M' of M by $\Psi(M')$ we denote $\cup_{t \in V(M')} \Psi(t)$.

Let (M, Ψ) be a tree decomposition of a graph G. The *width* of (M, Ψ) is $min\{|\Psi(t)| - 1 \mid t \in V(M)\}$, and the *adhesion* of the tree decomposition is $max\{|\sigma(t)| \mid t \in V(M)\}$. We use $\mathbf{tw}(G)$ to denote the treewidth of the input graph. If the For every node $t \in V(M)$, the *torso* at t is the graph

$$\tau(t) := G[\Psi(t)] \cup E(K[\sigma(t)]) \cup \bigcup_{u \, \text{child of} \, t} E(K[\sigma(u)]).$$

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 $k \leq |x|^{\mathcal{O}(1)}$. 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) $|x'| \leq g(k)$ and $k' \leq g(k)$, 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.

Known Decomposition Theorems. We start by the definition of nearly embeddable graphs.

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 decomposition theorem that we use extensively for our proofs is given in the next theorem.

Theorem 1 ([39, 47])**.** *For every graph H, there exists a constant h, depending only on the size of H, such that every graph G with $H \not\preceq_T G$, there is a tree decomposition (M, Ψ) of adhesion at most h such that for all $t \in V(M)$, one of the following conditions is satisfied:*

1. *$\tau(t)$ is h-nearly embedded in a surface Σ in which H cannot be embedded.*
2. *$\tau(t)$ has at most h vertices of degree larger than h.*

Moreover, if G is H-minor graph G then nodes of second type do not exist. Furthermore, there is an algorithm that, given graphs G, H of sizes n and $|H|$, computes such a tree decomposition in time $f(|H|)n^{O(1)}$ for some computable function f, and moreover computes an apex set Z_t of size at most h for every bag of the first type.

Furthermore we can assume that in (M, Ψ), for any $x, y \in V(M)$, $\Psi(x) \not\subseteq \Psi(y)$. That is, no bag is contained in other. See [28, Lemma 11.9] for the proof.

3 An approximation algorithm for DS on $H \not\preceq_T G$

In this section we give a constant factor approximation for DS on $\mathcal{C}_H$. It is well known that graphs in $\mathcal{C}_H$ has bounded degeneracy. In a recent manuscript a subset of the authors together with others show that DS has a $O(d^2)$ factor approximation algorithm on d-degenerate graphs [43]. To make this paper self contained we provide an approximation algorithm for DS on $\mathcal{C}_H$ here. For technical reasons, it is more convenient to give an approximation algorithm for a more general problem, namely Colored Dominating Set. In Colored Dominating Set, the vertex set of the input graph G is partitioned into X, Y, Z, $N(X) = Y$, and the objective is to find a minimum sized set $D \subseteq Y \cup Z$ that dominates all the vertices of Z. In other words, X is the set of vertices already in the dominating set, Y is the set of vertices already dominated, and the objective is to dominate Z, i.e. to find a minimum sized set $D \subseteq Y \cup Z$ such that $X \cup D$ is a dominating set for G. A normal Dominating Set instance G can be viewed as an instance of Colored Dominating Set by taking $X = Y = \emptyset$ and $Z = V(G)$.

Lemma 1. *Let H be a graph. Then there exists a constant $\eta(H)$ depending only on $|H|$ such that* Colored Dominating Set, *admits a $\eta(H)$-factor approximation algorithm on $\mathcal{C}_H$.*

Proof. We first describe the algorithm. Let G be the input with the vertex set partitioned into (X, Y, Z). During the algorithm we will be changing the input partition (X, Y, Z) but the solution for the resulting partition will be within a constant factor of the optimal solution for (X, Y, Z). We repeat the procedure described below until $Z \neq \emptyset$. The algorithm starts by computing the Grohe-Marx decomposition using Theorem 1. Let this tree decomposition be (M, Ψ). We root the tree M arbitrarily at some node w. We know that for every vertex $v \in V(G)$ we have that the nodes in the set $\{t \in V(M) \mid v \in \Psi(t)\}$ form a subtree M_v of M. A node $t \in M_v$ is the *peak* of a vertex v if t is the closest node to w in the rooted tree M. Select a lowermost node t that is the peak of an undominated vertex. In other words, t is the lowest node containing a vertex $v \in Z$ that has to be dominated and $v \notin \sigma(t)$. Now we have different procedure based on the type of the node t.

Case 1: *Torso $\tau(t)$ has at most h vertices of degree larger than h.* Let W be the set of vertices in $\tau(t)$ of degree more than h. Let $X^* = W \cup \sigma(t)$. Let $Z^* = (\Psi(t) \setminus N_G(X^*)) \cap Z$. We change partition (X, Y, Z) by adding $X^* \cup Z^*$ to X, making $Y = N(X)$, and decreasing $Z := V(G) \setminus (X \cup Y) := Z \setminus Y$.

Case 2: *Torso $\tau(t)$ is h-nearly embedded in a surface Σ in which H cannot be embedded.* In this case we first apply a PTAS for Colored Dominating Set developed in [19] for H-minor free graphs on $G[\Psi(t)]$ with $\Psi(t)$ being partitioned as follows: $X' = X \cap \Psi(t)$, $Y' = Y \cap \Psi(t)$ and $Z' = Z \cap \Psi(t)$. We run this PTAS as a factor two approximation algorithm. Let D be the set returned by this algorithm. That is, D dominates all the vertices in Z' and the size of D is at most twice the size of an optimal dominating set for Colored Dominating Set when the input is $G[\Psi(t)]$. We add $\sigma(t) \cup D$ to X. Finally, we make $Y := N(X)$ and $Z := V(G) \setminus (X \cup Y) := Z \setminus Y$.

Clearly, this algorithm runs in polynomial time. Next we show that this is indeed a constant factor approximation algorithm for Colored Dominating Set. We fix $\eta(H) = 5h$. We prove that the algorithm described above is a $\eta(H)$-factor approximation algorithm using induction on $p = |Z|$, the size of Z. The base case is when the input graph G has vertex set partitioned into X, Y and Z and $p = |Z| \leq 1$. The base case is obvious. For the induction case assume that the input graph G has vertex set partitioned into X, Y, and Z, $|Z| \geq 2$. This implies that either Case 1 or Case 2 applies. Let OPT be an optimum dominating set for G. By our choice of t we have that there exists a non-dominated vertex in $Z \cap \Psi(t)$ that is not contained in $\sigma(t)$ and all its neighbors in G are in $\gamma(t)$. This implies that $OPT \cap \gamma(t) \neq \emptyset$. Let $OPT_1 = OPT \cap \gamma(t)$.

Suppose that Case 1 applies. Let $OPT_1 \cap X^* = OPT_1'$. Then clearly $OPT_1 \setminus OPT_1'$ dominates all the vertices of Z^*. However all the vertices in $OPT_1 \setminus OPT_1'$ can dominate at most h vertices of Z^* and thus $(h+1)|OPT_1 \setminus OPT_1'| \geq |Z^*|$. This implies that $|X^*| + |Z^*| \leq 2h + (h+1)|OPT_1 \setminus OPT_1'| \leq (3h+1)|OPT_1|$. Observe that $OPT \setminus OPT_1$ is still a dominating set for the graph after we have updated X, Y and Z. However, the new set Z has been decreased as all the vertices in $\tau(t)$ are dominated now. Thus by appling the induction hypothesis, we obtain a dominating set of size at most $5h|OPT \setminus OPT_1|$. This implies that the total size of the dominating set returned by the algorithm in this case is at most $5h|OPT|$.

The proof for the Case 2 is similar. Observe that every vertex not present in $\Psi(t)$ can dominate only h vertices of $\Psi(t)$. Now we show that there exists a dominating set D' of $G[\Psi(t)]$ of size at most $h|OPT_1| + h$. We construct D' as follows. We add all the vertices in $OPT_1 \cap \Psi(t)$ and $\sigma(t)$ to D'. Moreover for every vertex in OPT_1 that is not in $\Psi(t)$, we add all its neighbors in $\Psi(t)$ to D'. Clearly the size of D' is at most $h|OPT_1| + h$ and D' is a dominating set of $G[\Psi(t)]$.

This implies that the set D returned by the algorithm is of size at most $2h|OPT_1| + 2h$. Hence $|D \cup \sigma(t)| \leq 2h|OPT_1| + 3h$. Observe that $OPT \setminus OPT_1$ is still a dominating set for the graph after we have updated X, Y and Z. However, the new set Z is smaller, as all the vertices in the $\tau(t)$ are dominated now. Thus by induction hypothesis, we can obtain a dominating set with size at most $5h|OPT \setminus OPT_1|$. This implies that the total size of the dominating set returned by the algorithm does not exceed $5h|OPT|$. This completes the proof. □

For CDS we need the following proposition attributed to [25].

Proposition 1. *Let G be a connected graph and let Q be a dominating set of G such that $G[Q]$ has at most ρ connected components. Then there exists a set $Z \subseteq V(G)$ of size at most $2 \cdot (\rho - 1)$ such that $Q \cup Z$ is a connected dominating set in G.*

Combining Lemma 1 and Proposition 1 we arrive to the following.

Lemma 2. *Let H be a graph and $\eta(H)$ the constant from Lemma 1. Then* CDS *admits a $3\eta(H)$-factor approximation algorithm on $\mathcal{C}_H$.*

4 Generalized Protrusions

A parameterized graph problem Π can be seen as a subset of $\Sigma^* \times \mathbb{Z}^+$ where, in each instance (x, k) of Π, x encodes a graph and k is the parameter (we denote by $\mathbb{Z}^+$ the set of all non-negative integers). Here we define the notion of *t-boundaried graphs* and various operations on them.

Definition 1. **[t-Boundaried Graphs]** *A t-boundaried graph is a graph G with a set $B \subseteq V(G)$ of at most t distinguished vertices and an injective labeling from B to the set $\{1, \ldots, t\}$,. The set B is called the* boundary *of G and vertices in B are called* boundary vertices *or* terminals. *Given a t-boundaried graph G we denote its boundary by $\delta(G)$. We use the notation $\mathcal{F}_t$ to denote the class of al t-boundaried graphs.*

We remark that in the labeling of the boundary of a t-boundaried graph, not all t available labels are necessary used.

Definition 2. **[Gluing by $\oplus$]** *Let G_1 and G_2 be two t-boundaried graphs. We denote by $G_1 \oplus G_2$ the graph obtained by taking the disjoint union of G_1 and G_2 and identifying equally-labeled vertices of the boundaries of G_1 and G_2. In $G_1 \oplus G_2$ there is an edge between two labeled vertices if there is an edge between them in G_1 or in G_2. When we are dealing with a gluing operation we use the term* common boundary *in G_1 and G_2 in order to denote the set of identified vertices in $G_1 \oplus G_2$.*

Definition 3. **[Gluing by $\oplus_\delta$]** *The* boundaried gluing operation $\oplus_\delta$ *is similar to the normal gluing operation, but results in a t-boundaried graph rather than a graph. Specifically $G_1 \oplus_\delta G_2$ results in a t-boundaried graph where the graph is $G = G_1 \oplus G_2$ and a vertex is in the boundary of G if it was in the boundary of G_1 or G_2. Vertices in the boundary of G keep their label from G_1 or G_2.*

Let $\mathcal{G}$ be a class of (not boundaried) graphs. By slightly abusing notation we say that a boundaried graph *belongs in a graph class $\mathcal{G}$* if the underlying graph belongs in $\mathcal{G}$.

Definition 4. **[Replacement]** *Let G be a t-boundaried graph containing a set $X \subseteq V(G)$ such that $\partial_G(X) = \delta(G)$. Let G_1 be a t-boundaried graph. The result of* replacing X with G_1 *is the graph $G^\star \oplus G_1$, where $G^\star = G \setminus (X \setminus \partial(X))$ is treated as a t-boundaried graph, where $\delta(G^\star) = \delta(G)$.*

4.1 Finite Integer Index

Definition 5. [**Equivalence of t-boundaried graphs**] *Let* Π *be a parameterized graph problem whose instances are pairs of the form* (G,k)*. Given two* t*-boundaried graphs* G_1, G_2*, we say that* $G_1 \equiv_{\Pi,t} G_2$ *if there exist a* transposition constant $c \in \mathbb{Z}$ *such that*

$$\forall (F,k) \in \mathcal{F}_t \times \mathbb{Z} \qquad (G_1 \oplus F, k) \in \Pi \Leftrightarrow (G_2 \oplus F, k+c) \in \Pi.$$

Note that for every t, the relation $\equiv_{\Pi,t}$ on t-boundaried graphs is an equivalence relation.

Next we define a notion of "transposition-minimality" for the members of each equivalence class of $\equiv_{\Pi,t}$.

Definition 6. [**Progressive representatives**] *Let* Π *be a parameterized graph problem whose instances are pairs of the form* (G,k) *and let* $\mathcal{C}$ *be some equivalence class of* $\equiv_{\Pi,t}$ *for some* $t \in \mathbb{Z}^+$*. We say that* $J \in \mathcal{C}$ *is a* progressive representative *of* $\mathcal{C}$ *if for every* $H \in \mathcal{C}$ *there exist* $c \in \mathbb{Z}^-$*, such that*

$$\forall (F,k) \in \mathcal{F}_t \times \mathbb{Z}\ (H \oplus F, k) \in \Pi \Leftrightarrow (J \oplus F, k+c) \in \Pi. \tag{1}$$

The following lemma guaranties the existence of a progressive representative for each equivalence class of $\equiv_{\Pi,t}$.

Lemma 3 ([9]). *Let* Π *be a parameterized graph problem whose instances are pairs of the form* (G,k) *and let* $t \in \mathbb{Z}^+$*. Then each equivalence class of* $\equiv_{\Pi,t}$ *has a progressive representative.*

Proof. Let $\mathcal{C}$ be an equivalence class of $\equiv_{\Pi,t}$. We distinguish two cases:

Case 1. Suppose first that for every $H \in \mathcal{C}$, every $F \in \mathcal{F}_t$, and every integer $k \in \mathbb{Z}^+$ it holds that $(H \oplus F, k) \notin \Pi$. Then we set J to be an arbitrary chosen graph in $\mathcal{C}$ and $c^* = 0$. In this case, it is obvious that (1) holds for every $(F,k) \in \mathcal{F}_t \times \mathbb{Z}$.

Case 2. Suppose now that for some $H_0 \in \mathcal{C}$, there exist an $F_0 \in \mathcal{F}_t$ and a non-negative integer k_0 such that $(H_0 \oplus F_0, k_0) \in \Pi$. Moreover choose k_0 as the minimum non-negative integer for which $(H_0 \oplus F_0, k_0) \in \Pi$.

Let $H \in \mathcal{C}$. As $H_0 \equiv_{\Pi,t} H$, there is a constant c^* such that

$$\forall (F,k) \in \mathcal{F}_t \times \mathbb{Z}\ (H_0 \oplus F, k) \in \Pi \Leftrightarrow (H \oplus F, k+c^*) \in \Pi \tag{2}$$

and we denote by $c(H)$ the minimum such constant c^*. To see that $c(H)$ is well defined, we claim that no such a c^* can be smaller than $-k_0$. Indeed, suppose that (2) holds for some c^*. When (2) is can be applied for (F_0, k_0) and along with the fact that $(H_0 \oplus F_0, k_0) \in \Pi$, it implies that $(H \oplus F_0, k_0 + c^*) \in \Pi$, where $k_0 + c^*$ is a parameter that is always a nonnegative integer. We conclude that $c^* \geq -k_0$ as claimed.

By the choice of k_0, we obtain $c(H) = 0$. We define J to be a member of $\mathcal{C}$ where

$$c(J) \quad = \quad \min\{c(H) \mid H \in \mathcal{C}\}. \tag{3}$$

From 3, we obtain that $c(J) \leq c(H)$, for every $H \in \mathcal{C}$. Therefore, the fact that $c(H_0) = 0$ implies that $c(J) \leq 0$. We set $c' = c(J)$.

Let $H \in \mathcal{C}$. We set $c'' = c(H)$ and notice that $c' - c'' \leq 0$. From $H_0 \equiv_{\Pi,t} H$ and the definition of $c(H)$,

$$\forall (F,k) \in \mathcal{F}_t \times \mathbb{Z}\ (H_0 \oplus F, k) \in \Pi \Leftrightarrow (H \oplus F, k+c'') \in \Pi \tag{4}$$

Similarly, $H_0 \equiv J$ implies that

$$\forall (F,k) \in \mathcal{F}_t \times \mathbb{Z}\ (H_0 \oplus F, k) \in \Pi \Leftrightarrow (J \oplus F, k+c') \in \Pi \tag{5}$$

We rewrite (4) and (5) as follows:

$$\forall (F,k) \in \mathcal{F}_t \times \mathbb{Z}\ (H \oplus F, k) \in \Pi \quad \Leftrightarrow \quad (H_0 \oplus F, k - c'') \in \Pi \tag{6}$$

$$\forall (F,k) \in \mathcal{F}_t \times \mathbb{Z}\ (H_0 \oplus F, k - c'') \in \Pi \quad \Leftrightarrow \quad (J \oplus F, k + c' - c'') \in \Pi \tag{7}$$

It follows from (6) and (7) that

$$\forall (F,k) \in \mathcal{F}_t \times \mathbb{Z}\ (H \oplus F, k) \in \Pi \Leftrightarrow (J \oplus F, k + c' - c'') \in \Pi \tag{8}$$

Clearly (8) together with the fact that $c' - c'' \in \mathbb{Z}^-$ implies that (1) holds if we set $c = c' - c''$ as required. □

After Lemma 3 we are in position to give the following definition.

Definition 7. *A parameterized graph problem* Π *whose instances are pairs of the form* (G,k) *has* Finite Integer Index *(or simply has* FII*), if and only if for every* $t \in \mathbb{Z}^+$*, the equivalence relation* $\equiv_{\Pi,t}$ *is of finite index, that is, has a finite number of equivalence classes. For each* $t \in \mathbb{Z}^+$*, we define* $\mathcal{S}_t$ *to be a set containing exactly one progressive representative of each equivalence class of* $\equiv_{\Pi,t}$*.*

Definition 8. *We say that a parameterized graph problem* Π *is* positive monotone *if for every graph* G *there exists a unique* $\ell \in \mathbb{N}$ *such that for all* $\ell' \in \mathbb{N}$ *and* $\ell' \geq \ell$*,* $(G,\ell') \in \Pi$ *and for all* $\ell' \in \mathbb{N}$ *and* $\ell' < \ell$*,* $(G,\ell') \notin \Pi$*. A parameterized graph problem* Π *is* negative monotone *if for every graph* G *there exists a unique* $\ell \in \mathbb{N}$ *such that for all* $\ell' \in \mathbb{N}$ *and* $\ell' \geq \ell$*,* $(G,\ell') \notin \Pi$ *and for all* $\ell' \in \mathbb{N}$ *and* $\ell' < \ell$*,* $(G,\ell') \in \Pi$*.* Π *is monotone if it is either positive monotone or negative monotone. We denote the integer* ℓ *by* $\textsc{Thr}(G)$*.*

Definition 9. *Let* Π *be a monotone parameterized graph problem that is FII. Let* $\mathcal{S}_t$ *to be a set containing exactly one progressive representative of each equivalence class of* $\equiv_{\Pi,t}$*. For a* t*-boundaried graph* G *by* $\kappa(G)$ *we denote*

$$\max_{G' \in \mathcal{S}_t} \textsc{Thr}(G \oplus G').$$

Lemma 4. *Let* Π *be a monotone parameterized graph problem that is FII. Furthermore, let* $\mathcal{A}$ *be an algorithm for* Π *that given a pair* (G,k) *decides whether it is in* Π *in time* $f(|V(G)|, k)$*. Then for every* $t \in \mathbb{N}$*, there exists a* $\xi_t \in \mathbb{Z}^+$ *(depending on* Π *and* t*), and an algorithm that, given a* t*-boundaried graph* G *with* $|V(G)| > \xi_t$*, outputs, in* $O(\kappa(G)(f(|V(G)| + \xi_t, \kappa(G)))$ *steps, a* t*-boundaried graph* G^* *such that* $G \equiv_{\Pi,t} G^*$ *and* $|V(G^*)| < \xi_t$*. Moreover we can compute the translation constant* c *from* G *to* G^* *in the same time.*

We remark that the algorithm whose existence is guaranteed by the Lemma 4 assumes that the set $\mathcal{S}_t$ of representatives are hardwired in the algorithm and that in general there is no procedure that for FII problems Π outputs such a representative set.

Proof. We give proof only for positive monotone problem Π, the proof for negative monotone is identical. Recall that we denote by $\mathcal{S}_t$ a set of (progressive) representatives for (Π, t) and let $\xi_t = \max_{Y \in \mathcal{S}_t} |Y|$. The set $\mathcal{S}_t$ is hardwired in the code of the algorithm. Let $Y_1, \ldots, Y_\rho$ be the set of progressive representatives in $\mathcal{S}_t$. Our objective is to find a representative Y_ℓ for G such that

$$\forall (F,k) \in \mathcal{F}_t \times \mathbb{Z} \qquad (G \oplus F, k) \in \Pi \Leftrightarrow (Y_\ell \oplus F, k - \eta(X, Y_\ell)) \in \Pi. \tag{9}$$

Here, $\eta(X, Y_\ell)$ is a constant that depends on G and Y_ℓ. Towards this we make the following matrix for the set of representatives. Let

$$A[Y_i, Y_j] = \textsc{Thr}(Y_i \oplus Y_j)$$

The matrix A has constant size and is also hardwired in the code of the algorithm.

Now given G we find its representative as follows.

- Compute the following row vector $\mathcal{X} = [\textsc{Thr}(G \oplus Y_1), \textsc{Thr}(G \oplus Y_2), \ldots, \textsc{Thr}(G \oplus Y_\rho))]$. For each Y_i we decide whether $(G \oplus Y_i, k) \in \Pi$ using the assumed algorithm for deciding Π, letting k increase from 1 until the first time $(G \oplus Y_i, k) \in \Pi$. Since Π is positive monotone this will happen for some $k \leq \kappa(G)$. Thus the total time to compute the $\mathcal{X}$ is $O(\kappa(G)(f(|V(G)| + \xi_t, \kappa(G)))$.
- Find a translate row in the matrix $A(\Pi)$. That is, find an integer n_o such that there exists a representative Y_ℓ such that

$$\begin{aligned}&[\textsc{Thr}(G \oplus Y_1), \textsc{Thr}(G \oplus Y_2), \ldots, \textsc{Thr}(G \oplus Y_\rho))]\\ = &[\textsc{Thr}(Y_\ell \oplus Y_1) + n_0, \textsc{Thr}(Y_\ell \oplus Y_2) + n_0, \ldots, \textsc{Thr}(Y_\ell \oplus Y_\rho)) + n_0]\end{aligned}$$

  Such a row must exist since $\mathcal{S}_t$ is a set of representatives for Π.
- Set Y_ℓ to be G^* and the translation constant to be $-n_0$.

From here it easily follows that $G \equiv_{\Pi,t} G^*$. This completes the proof. □

5 Slice-Decomposition

In this section our objective is to show that in polynomial time we can partition the graph G satisfying certain properties. To obtain our decomposition we need to use a more general notion of protrusion. More precisely, we need the following kind of protrusions.

Definition 10. [r-DS**-protrusion**] *Given a graph G, we say that a set $X \subseteq V(G)$ is an* r-DS-protrusion *of G if the number of vertices in X with a neighbor in $V(G) \setminus X$ is at most r and there exists a subset $S \subseteq X$ of size at most r such that X is a dominating set of $G[X]$.*

The notion of r-DS-protrusion X differs from normal protrusion in the following way. In the normal protrusion we demand that $\mathbf{tw}(X)$ is at most r while in the r-DS-protrusion we demand that the dominating set of the graph induced on X is small. We can similarly define the notion of r-Π-protrusion for various other graph problems Π.

Definition 11. [r-CDS**-protrusion**] *Given a graph G, we say that a set $X \subseteq V(G)$ is an* r-CDS-protrusion *of G if the number of vertices in X with a neighbor in $V(G) \setminus X$ is at most r and there exists a subset $S \subseteq X$ of size at most r such that for every connected component C of $G[X]$ we have that $X \cap C$ is a connected dominating set for C.*

The next question is what do we achieve if we get a large r-DS-protrusion (or r-CDS-protrusion). The next lemma shows that in that case we can replace it with an equivalent small graph. More precisely we have the following.

Lemma 5. *Let H be a fixed graph. For very $t \in \mathbb{Z}^+$, there exist a $\xi_t \in \mathbb{Z}^+$ (depending on* DS *(*CDS*), t and H), and an algorithm $\mathcal{A}$ such that given a t-*DS*-protrusion (t-*CDS*-protrusion) with $|X| > \xi_t$, and $H \not\preceq_T X$, $\mathcal{A}$ outputs in $\mathcal{O}(|X|)$ time ($|X|^{\mathcal{O}(1)}$ time), a t-boundaried graph X' such that $X \equiv_{\mathrm{DS},t} X'$ ($X \equiv_{\mathrm{CDS},t} X'$) and $|X'| \leq \xi_t$. Moreover in the same time we can also find the translation constant c from X to X'.*

Proof. For every $t \in \mathbb{Z}^+$ let ξ_t be the constant as defined in Lemma 4. It is also known that both DS (CDS) are FII [9] and monotone. Furthermore, DS and CDS can be solved in time $\mathcal{O}(hk)^{hk}n$ [4, Theorem 4] and $\mathcal{O}(k^{\mathcal{O}(h^2)k}n^{\mathcal{O}(1)})$ [37, Theorem 1] respectively. Thus, if $|X| > \xi_t$ then by Lemma 4 in time $\mathcal{O}(|X|)$ ($|X|^{\mathcal{O}(1)}$)), we can obtain a t-boundaried graph X' such that $X \equiv_{\mathrm{DS},t} X'$ ($X \equiv_{\mathrm{CDS},t} X'$) and $|X'| < \xi_t$. Moreover in the same time we can also find the translation constant c from X to X' as done in Lemma 4. □

1. Apply Lemma 1 (Lemma 2) on the input graph G and compute a (connected) dominating set D such that the size of D is at most $\eta(H)$-factor away from the size of an optimal dominating set of G.
2. Use Theorem 1 and compute a tree-decomposition (M, Ψ). We call a tree edge $e = uv \in E(M)$ *heavy* if $\mu(M_u, D) \geq h+1$ and $\mu(M_v, D) \geq h+1$. Mark all the edges of M that are heavy. We use $\mathcal{F}$ to denote the set of edges that have been marked.

Figure 2: Marking heavy edges.

Let (M, Ψ) be a tree decomposition of a graph G. For a subtree M_i of M, we define $\mathcal{E}(M_i)$ as the set of edges in M such that it has exactly one endpoint in $V(M_i)$. Furthermore we define $R_i^+ = \Psi(M_i)$ and

$$\tau(M') := G[R_i^+]) \cup \bigcup_{e \in \mathcal{E}(M_i)} E(K[\kappa(e)]).$$

Our main objective in this section is to obtain the following (α, β)-*slice decomposition* for $\alpha = \beta = \mathcal{O}(k)$.

Definition 12. [**(α, β)-slice decomposition**] *Let G be a graph with $H \not\preceq_T G$ and let (M, Ψ) be the tree decomposition given by Theorem 1. An (α, β)-*slice decomposition *of a graph G is a collection $\mathcal{P}$ of pairwise disjoint connected subtrees $\{M_1, \ldots, M_\alpha\}$ of M such that the following holds.*

- *Each of $\tau(M_i)$ is either H^*-minor free for some graph H^* whose size only depends on h or $\tau(M_i)$ has at most h vertices of degree at least h.*
- *$\sum_{i=1}^{\rho} (\sum_{e \in \mathcal{E}(M_i)} |\kappa(e)|) \leq \beta$.*

We call the sets R_i^+, $i \in \{1, \ldots, \rho\}$, slices *of $\mathcal{P}$.*

Essentially, the slice-decomposition allows us to partition the input graph G into subgraphs $C_0, C_1, \ldots, C_\ell$, such that $|C_0| = \mathcal{O}(k)$; for every $i \geq 1$, the neighbourhood $N(C_i) \subseteq C_0$, and $\sum_{1 \leq i \leq \ell} |N(C_i)| = \mathcal{O}(k)$. Now we define a notion of measure.

Definition 13. *Let (M, Ψ) be the tree decomposition of a graph G given by the Theorem 1. For a subset $Q \subseteq V(G)$ and a subtree M' of M we define $\mu(M', Q) = |\Psi(M') \cap Q|$. If we delete an edge $e = uv \in E(M)$ from the tree M then we get two trees. We call the trees as M_u and M_v based on whether they contain u or v.*

Our main lemma in this section shows that in polynomial time either we find a $(\mathcal{O}(k), \mathcal{O}(k))$-slice decomposition or a large r-DS-protrusion (or r-CDS-protrusion) or a normal protrusion. In the later cases we can apply either Lemma 5 or a similar lemma developed in [9, Lemma 7] for normal protrusions and reduce the graph. Towards the proof of our main lemma we first introduce a marking scheme (see Figure 2).

Before we prove the main result of this section, we prove some combinatorial properties of the marking schema described in Figure 2 that will be useful later.

Lemma 6. *Let M^* be the subgraph formed by the edges in $\mathcal{F}$ then M^* is a subtree of M.*

Proof. Clearly, M^* is a forest as it is a subgraph of M. To complete the proof we need to show that it is connected. We prove this using contradiction. Suppose there are two trees M_i^* and

M_j^*, $i \neq j$. Then we know that there exists a path P such that the first and the last edges are heavy and the path P contains a light edge. Furthermore, we can assume that the first edge, say u_iv_i, belongs to M_i and the last edge, say u_jv_j belongs to M_j. Let the light edge on the path be xy. Now when we delete the edge xy from M we get two trees M_x and M_y. We know that either $M_i^* \subseteq M_x$ and $M_j^* \subseteq M_y$ or vice versa. Suppose $M_i^* \subseteq M_x$ and $M_j^* \subseteq M_y$. Since M_i^* contains the heavy edge u_iv_i we have that $\mu(M_x, D) \geq h+1$. Similarly we can show that $\mu(M_y, D) \geq h+1$. This shows that xy is a heavy edge and hence would have been marked. One can similarly argue that xy is a heavy edge when $M_i^* \subseteq M_y$ and $M_j^* \subseteq M_x$. This is contradiction to our assumption that M^* is not a subtree of M. This completes the proof of the lemma. □

For our next proof we first give some well known observations about trees. Given a tree T, we call a node *leaf*, *link* or *branch* if its degree in T is 1, 2 or ≥ 3 respectively. Let $S_{\geq 3}(T)$ be the set of branch nodes, $S_2(T)$ be the set of link nodes and $L(T)$ be the set of leaves in the tree T. Let $\mathscr{P}_2(T)$ be the set of maximal paths consisting of link nodes.

Fact 1. *$|S_{\geq 3}(T)| \leq |L(T)| - 1$.*

Fact 2. *$|\mathscr{P}_2(T)| \leq 2|L(T)| - 1$.*

Proof. Root the tree at an arbitrary node of degree at least 3. If there is no node of degree 3 or more in T then we know that the T is a path and the assertion follows. Consider $T[S_2]$ which is the disjoint union of paths $P \in \mathscr{P}_2(T)$. With every path $P \in \mathscr{P}_2(T)$, we associate the unique child of the last node of this path in T. Observe that this association is injective and the associated node is either a leaf or a branch node. Hence

$$|\mathscr{P}_2(T)| \leq |L(T)| + |S_{\geq 3}(T)| \leq 2|L(T)| - 1$$

from Fact 1. □

Lemma 7. *If (G, k) is a yes instance to* DS *(*CDS*) then (a) $|L(M^*)| \leq \eta(H)k$; (b) $|S_{\geq 3}(M^*)| \leq \eta(H)k - 1$; and (c) $|\mathscr{P}_2(M^*)| \leq 2\eta(H)k - 1$. Here $\eta(H)$ is the factor of approximation in Lemma 1 (Lemma 2).*

Proof. Root the tree at an arbitrary node of degree at least 3, say r. If there is no node of degree 3 or more in M^* then we know that the T is a path and the proof follows. We call a pair of nodes u and v *siblings* if they do not belong to the same path from the root r in M^*. Observe that all the leaves of M^* are siblings.

Let $w_1, \ldots, w_\ell$ be the leaves of M^* and let $e_1, \ldots, e_\ell$ be the corresponding edges incident to $w_1, \ldots, w_\ell$, respectively. To prove our first statement we will show that for every i, we have a vertex $q_i \in D$ such that $q_i \in \gamma(w_i)$ and for all $j \neq i$, $q_i \notin \gamma(w_j)$. This will establish an injection from the set of leaves to the dominating set D and thus the bound will follow. Towards this observe that since the edge e_i is marked we have that $|\gamma(w_i) \cap D| \geq h+1$. Furthermore, for every pair of vertices $w_i, w_j \in L(M^*)$, $w_i \neq w_j$, we have that $|\gamma(w_i) \cap \gamma(w_j)| \leq h$. The last assertion follows from the fact that for a pair of siblings w_i and w_j the only vertices that can be in the intersection of $\gamma(w_i)$ and $\gamma(w_j)$ must belong to both $\sigma(w_i)$ and $\sigma(w_j)$. However, the size of any $\sigma(w_i)$ is upper bounded by h. This together with the fact that $|\gamma(w_i) \cap D| \geq h+1$ implies that for every i, we have a vertex $q_i \in D$ such that $q_i \in \gamma(w_i)$ and for all $j \neq i$, $q_i \notin \gamma(w_j)$. This implies that $|L(M^*)| \leq |D|$. However since (G, k) is a yes instance to DS we have that $|D| \leq \eta(H)k$. This completes the proof of part (a) of the lemma. Proofs for part (b) and part (c) of the lemma follow from Facts 1 and 2. □

Before we prove our next lemma we show a lemma about dominating set of subgraphs of G.

Lemma 8. *Let G be a graph such that $H \not\preceq_T G$ and (M, Ψ) be the tree decomposition given by Theorem 1. Let M' be a subtree of M then $(D \cap \Psi(M')) \cup_{e \in \mathcal{E}(M')} \kappa(e)$ is a dominating set for $G[\Psi(M')]$.*

Proof. The proof follows from the fact that $D \cap \Psi(M')$ dominates all the vertices in $\Psi(M')$ except the ones that have neighbors in $V(G) \setminus (\cup_{e \in \mathcal{E}(M')} \kappa(e))$. Thus, $(D \cap \Psi(M')) \cup_{e \in \mathcal{E}(M')} \kappa(e)$ is a dominating set for $G[\Psi(M')]$. □

Let $P_1, \ldots, P_l$ be the paths in $\mathscr{P}_2(M^*)$. We use s_i and t_i to denote the first and the last vertex of the path P_i. Since P_i is a path consisting of link vertices we have that s_i and t_i have unique neighbors s_i^* and t_i^* respectively in M^*. Observe that since M^* is a subtree of M, we have that for every i, P_i is also a path in M. If we delete the edges $s_i^* s_i$ and $t_i^* t_i$ from the tree M, we get a subtree that contains the path P_i, we call this subtree $M(P_i)$. For any two vertices a and b on the path P_i by $P_i(a, b)$ we denote the subpath between a and b in P_i. Furthermore for any subpath $P_i(a, b)$, if we delete the edges incident to a and b on P_i and not present in $P_i(a, b)$ from the tree M, we get a subtree that contains the path $P_i(a, b)$, we call this subtree $M(P_i(a, b))$. Now we are ready to state our next lemma.

Lemma 9. *Let (G, k) be an instance to* DS*(*CDS*). Then, if for any path P_i, $i \in \{1, \ldots, \ell\}$, we have that $|P_i| > \xi_{2h} 2(2h + k_i)$ then G contains a $2h$-*DS*-protrusion of size at least ξ_{2h}. Here, $k_i = |D \cap \Psi(M(P_i))|$.*

Proof. We prove this lemma using contradiction. Suppose for some $i \in \{1, \ldots, \ell\}$, we have that $|P_i| > 2\delta_2(|D \cap \Psi(M(P_i))|$. Let $P_i := s_i = a_1^i \cdots a_{t_i}^i = t_i$. For every vertex w in $D \cap \Psi(M(P_i))$ and $\sigma(s_i^*) \cup \sigma(t_i)$ we mark two vertices of the path P_i. We mark the first and the last vertices on P_i, say a_x^i and a_y^i, such that $w \in \Psi(M(P_i(a_x^i))$ and $w \in \Psi(M(P_i(a_y^i)))$. That is, $w \in \Psi(M(P_i(a_x^i))$ and $w \in \Psi(M(P_i(a_y^i))$ and for all $z < x$ or $z > y$ we have that $w \notin \Psi(M(P_i(a_z^i))$. This way we will only mark at most $2(2h + |D \cap \Psi(M(P_i))|) = 2(2h + k_i)$ vertices of the path P_i. However the path is longer than $2\xi_{2h}(2h + k_i)$ and thus by pigeonhole principle we have that there exists a subpath of P_i, say $P_i(a_x^i, a_y^i)$, such that no vertex of this subpath is marked and $|P_i(a_x^i, a_y^i)| > \xi_{2h}$. Let $W = \Psi(M(P_i(a_x^i, a_y^i)))$. Let a be the neighbor of a_x^i in M^* that is not present on $P_i(a_x^i, a_y^i)$. Clearly, the only vertices in W that have neighbors in $V(G) \setminus W$ belong to $\sigma(a) \cup \sigma(a_y^i)$. Thus it is upper bounded by $2h$. Furthermore, since no vertex on the path $P_i(a_x^i, a_y^i)$ is marked we have that all the vertices in D belonging to W also belongs to $\sigma(a) \cup \sigma(a_y^i)$. Thus by Lemma 8 we have that $\sigma(a) \cup \sigma(a_y^i)$ dominates all the vertices in W. Furthermore, in (M, Ψ), no bag is contained in other and thus $|W| > \xi_{2h}$. This shows that W is a $2h$-DS-protrusion of desired size. □

Lemma 10. *Let H be a fixed graph and $\mathcal{C}_H$ be the class of graphs excluding H as a topological minor. Then there exist two constants δ_1 and δ_2 (depending on* DS *(*CDS*)) such that given a yes instance (G, k) of* DS *(*CDS*), in polynomial time, we can either find*

- *$(\delta_1 k, \delta_2 k)$-slice decomposition; or*
- *a $2h$-*DS*-protrusion (or $2h$-*CDS*-protrusion) of size more than ξ_{2h} or;*
- *a h'-protrusion of size more than $\xi_{h'}$ where h' depends only on h.*

Proof. Let (G, k) be a yes instance of DS(CDS). This implies that the size of the (connected) dominating set D returned by Lemma 1(Lemma 2) is at most $\eta(H)k$. Now we apply the marking scheme as described in Figure 2. Let M^* be the subtree of M induced on heavy edges. By Lemma 7, we know that (a) $|L(M^*)| \leq \eta(H)k$; (b) $|S_{\geq 3}(M^*)| \leq \eta(H)k - 1$; and (c) $|\mathscr{P}_2(M^*)| \leq 2\eta(H)k - 1$. Recall that for every path $P_i \in \mathscr{P}_2(M^*)$, we defined $k_i = |D \cap \Psi(M(P_i))|$. If for any path $P_i \in \mathscr{P}_2(M^*)$ we have that $|P_i| > \xi_{2h} 2(2h + k_i)$ then by Lemma 9 G contains a

$2h$-DS-protrusion of size at least ξ_{2h}, and we can find this protrusion in polynomial time. Thus we assume that for all paths $P_i \in \mathscr{P}_2(M^*)$ we have that $|P_i| \leq \xi_{2h} 2(2h + k_i)$.

Let k_i^* denote the number of vertices in $D \cap \Psi(M(P_i))$ that are not present in any other $D \cap \Psi(M(P_j))$ for $i \neq j$. Furthermore, for all $i \neq j$ we have that $\left|\Big(D \cap \Psi(M(P_i))\Big) \cap \Big(D \cap \Psi(M(P_j)\Big)\right| \leq h$. Thus we have that $k_i \leq h + k_i^*$. This implies that

$$
\begin{aligned}
|V(M^*)| &= |L(M^*)| + |S_{\geq 3}| + |S_2| \\
&\leq \eta(H)k + \eta(H)k - 1 + \sum_{P_j \in \mathscr{P}_2(M^*)} (4h + 2k_j)\xi_{2h} \\
&\leq 2\eta(H)k - 1 + 4h|\mathscr{P}_2(M^*)|\xi_{2h} + \sum_{P_j \in \mathscr{P}_2(M^*)} 2(h + k_j^*)\xi_{2h} \\
&\leq 2\eta(H)k - 1 + 6h|\mathscr{P}_2(M^*)|\xi_{2h} + 2|D|\xi_{2h} \\
&\leq (2 + 12h\xi_{2h} + 2\xi_{2h})\eta(H)k
\end{aligned}
$$

This implies that the number of marked edges is upper bounded by $|\mathcal{F}| \leq (2 + 12h\xi_{2h} + 2\xi_{2h})\eta(H)k - 1$. Let $M_1, \ldots, M_\alpha$ be the subtrees of M obtained by deleting all the edges in M^*, that is, by deleting all the edges in $\mathcal{F}$. Note that α is upper bounded by $(2+12h\xi_{2h}+2\xi_{2h})\eta(H)k$. We now argue that the collection $M_1, \ldots, M_\alpha$ forms a $(\delta_1 k, \delta_2 k)$-slice decomposition of G or we will find a $2h$-protrusion or a $2h$-DS-protrusion of size more than ξ_{2h} in G.

First we show that $\sum_{i=1}^{\alpha}(\sum_{e \in \mathcal{E}(M_i)} |\kappa(e)|) = \mathcal{O}(k)$. Specifically since every heavy edge belongs to at most 2 distinct edge sets $\mathcal{E}(M_i)$, we have that

$$
\sum_{i=1}^{\alpha} \sum_{e \in \mathcal{E}(M_i)} |\kappa(e)| \leq 2 \sum_{e \in E(M^*) = \mathcal{F}} |\kappa(e)| \leq 2h|\mathcal{F}| \leq 2h((2 + 12h\xi_{2h} + 2\xi_{2h})\eta(H)k - 1).
$$

We set $\delta_2 = 2h(2 + 12h\xi_{2h} + 2\xi_{2h})\eta(H)$, and $\delta_1 = \frac{\alpha}{k}$, since $\alpha = O(k)$ we have that δ_1 is a constant.

Since M^* is connected we have that for every tree M_i there is a unique node that is adjacent to edges in $\mathcal{F}$. We denote this special node by r_i. We root the tree M_i at r_i. Let w be a child of r_i and let M_w and M_{r_i} be the trees of M obtained after deleting the edge $r_i w$. Since at least one edge incident to r_i is heavy we have that $\mu(M_{r_i}, D) \geq h+1$. However the edge $r_i w$ is not heavy and thus it must be the case that $\mu(M_w, D) \leq h$. Let $W = \Psi(M_w)$. Then by Lemma 8 we have that $(D \cap W) \cup \kappa(r_i w) = \sigma(w)$ is a dominating set of size at most $2h$ for $G[W]$. Furthermore, the only vertices in W that have neighbors in $V(G) \setminus W$ belong to $\sigma(w)$ and thus its size is also upper bounded by h. This implies that if $|W| > \xi_{2h}$ then it is a $2h$-DS-protrusion of size at least ξ_{2h}. Thus from now onwards we assume that this is not the case.

In the case when $\tau(r_i)$ has at most h vertices of degree larger than h, we show there exists an h' depending only on h such that either $\tau(M_i)$ has at most $h' = \xi_{h+\xi_{2h}}$ vertices of degree larger than h or G contains a $2h$-protrusion of size more than ξ_{2h}. Suppose some vertex v in $\tau(r_i)$ has degree at most h in $\tau(r_i)$, but has degree at least h' in $\tau(M_i)$. Let N be the closed neighbourhood of v in $\tau(r_i)$ and N' be the neighborhood of v in $\tau(M_i)$. Each vertex in $N' \setminus N$ must lie in a connected component C of $\tau(M_i) \setminus N$ on at most ξ_{2h} vertices. Furthermore, no vertex in C sees any vertex outside N even in the graph G. Let X be N plus the union of all such components. By assumption $|N' \setminus N| \geq \xi_{2h}$ and hence $|X| \geq \xi_{2h}$. Finally, the only vertices in X that have neighbors outside of X in G are in N, and $|N| \leq h$. The treewidth of $G[X]$ is at most $\xi_{2h} + h$ since removing N from X leaves components of size ξ_{2h}. Thus X is a h'-protrusion of size more than $\xi_{h'}$. If no such X exists it follows that every vertex of degree at most h in $\tau(r_i)$ has degree at most h' in $\tau(M_i)$. The vertices of $\tau(M_i)$ that are not in $\tau(r_i)$ have degree at most $\xi_{2h} + h < h'$. Thus $\tau(M_i)$ has at most $h < h'$ vertices of degree at least h'.

In the case that $\tau(r_i)$ is h-nearly embedded in a surface Σ in which H cannot be embedded, we have that $\tau(r_i)$ excludes some graph H' depending only on h as a minor. The graph $\tau(M_i)$ can be obtained from $\tau(r_i)$ by joining constant size graphs (of size at most ξ_{2h}) to vertex sets that form cliques in $\tau(r_i)$. Thus there exists a graph H^* depending only on h such that $\tau(M_i)$ excludes H^* as a minor. This completes the proof of this lemma. $\square$

6 Final Kernel

In this section we use slice-decomposition obtained in the last section and the reduction rules used in [34] to obtain linear kernels for DS and CDS. We first outline our algorithm for DS and then explain how we can obtain a linear kernel for CDS.

6.1 Kernelization Algorithm for DS

Given an instance (G, k) of DS we first apply Lemma 1 and find a dominating set D of G. If $|D| > \eta(H)k$ we return that (G, k) is a NO instance to DS. Else, we apply Lemma 10 and

- either find $(\delta_1 k, \delta_2 k)$-slice decomposition; or
- a $2h$-DS-protrusion X of G (or $2h$-CDS-protrusion) of size more than ξ_{2h}; or
- a h'-protrusion of size more than $\xi_{h'}$ where h' depends only on h.

In the second case we apply Lemma 5. Given X we apply Lemma 5 and obtain a boundaried graph X' such that $|X'| \leq \xi_{2h}$ and $X \equiv_{\mathrm{DS},2h} X'$ ($X \equiv_{\mathrm{CDS},2h} X'$). We also compute the translation constant c between X and X'. Now we replace the graph X with X' and obtain a new equivalent instance $(G', k + c)$. (Recall that c is a negative integer). In the third case we apply the protrusion replacement lemma of [9, Lemma 7] to obtain a new equivalent instance (G', k') for $k' \leq k$ with $|V(G')| < |V(G)|$. We repeat this process until Lemma 10 returns a slice-decomposition. For simplicity we denote by (G, k) itself the graph on which Lemma 10 returns the slice-decomposition. Since the number of times this process can be repeated is upper bounded by $n = |V(G)|$, we can obtain $(\delta_1 k, \delta_2 k)$-slice decomposition for (G, k) in polynomial time.

Let $\mathcal{P}$ be the pairwise disjoint connected subtrees $\{M_1, \ldots, M_\alpha\}$ of M coming from the slice-decomposition of G. Recall that $R_i^+ = \Psi(M_i)$. Let $Q_i = \bigcup_{e \in \mathcal{E}(M_i)} \kappa(e)$, $B_i = (D \cap R_i^+) \cup Q_i$ and $b_i = |B_i|$. In this section we will treat $G_i := G[R_i^+]$ as a graph with boundary B_i. Observe that by Lemma 8, it follows that B_i is a dominating set for G_i.

We have two kinds of graphs G_i. In one case we have that G_i is H^*-minor free for a graph H^* whose size only depends on h. In the other case we have that the graph G_i has at most h' vertices of degree at least h'. To obtain our kernel we will show the following two lemmas.

Lemma 11. *There exists a constant δ such that graph G with boundary S such that S is a dominating set for G and G has at most h' vertices of degree at least h', then in polynomial time, we can obtain a graph G' with boundary S such that*

$$G' \equiv_{\mathrm{DS},b_i} G \text{ and } |V(G')| \leq \delta|S|.$$

Furthermore we can also compute the translation constant c of G and G' in polynomial time.

Lemma 12. *There exists a constant δ such that given a H-minor free graph G with boundary S such that S is a dominating set for G, in polynomial time, we can obtain a graph G' with boundary S such that*

$$G' \equiv_{\mathrm{DS},b_i} G \text{ and } |V(G')| \leq \delta|S|.$$

Furthermore we can also compute the translation constant c of G and G' in polynomial time.

Once we have proved Lemmata 11 and 12, we obtain the linear sized kernel for DS as follows. Given the graph G we obtain the slice-decomposition and check if any of G_i has size more than δb_i. If yes then we either apply Lemma 11 or Lemma 12 based on the type of G_i and obtain a graph G'_i such that $G'_i \equiv_{\mathrm{DS},b_i} G_i$ and $|V(G'_i)| \leq \delta b_i$. We think $G = G_i \oplus G^\star$, where $G^\star = G \setminus (R_i^+ \setminus B_i)$ as a b_i-boundaried graph with boundary B_i. Then we obtain a smaller equivalent graph $G' = G^\star \oplus G'_i$ and $k' = k + c$. After this we can repeat the whole process once again. This implies that when we can not apply Lemmata 12 or 11 on (G, k) we have that each of $|V(G_i)| \leq \delta b_i$. Furthermore notice that $\cup_{i=1}^{\alpha} R_i^+ = V(G)$. This implies that in this case we have the following

$$\begin{aligned}
\sum_{i=1}^{\alpha} |R_i^+| &\leq \delta \sum_{i=1}^{\alpha} b_i = \delta(\sum_{i=1}^{\alpha}(|Q_i| + |(D \cap R_i^+) \setminus Q_i|)) \\
&= \delta(\sum_{i=1}^{\alpha} |Q_i| + \sum_{i=1}^{\alpha} |(D \cap R_i^+) \setminus Q_i|) \leq \delta\delta_2 k + \delta\eta(H)k = \mathcal{O}(k).
\end{aligned}$$

This brings us to the following theorem.

Theorem 2. DS *admits a linear kernel on graphs excluding a fixed graph H as a topological minor.*

It only remains to prove Lemmas 11 and 12 to complete the proof of Theorem 3.

6.2 Irrelevant Vertex Rule and proofs for Lemmas 11 and 12

For the proofs of Lemmas 11 and 12 we need to use an irrelevant vertex rule developed in [34]. Furthermore, the proof of Lemma 12 is essentially a reformulation of the results presented in [34].

If the graph G is $K_{h'}$-minor free then 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 the bottom of the recursion when the graph becomes smaller but still big enough then we apply Lemma 5 on it and obtain an equivalent instance.

Let G be a graph given with its tree-decomposition (M, Ψ) as described in Theorem 1, and $\tau(t)$ be one of its torsos. Let S be a dominating set of G, and $Z_t = A$, $|A| \leq h$, be the set of apices of $\tau(t)$. 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 $|S|+1$ such that D dominates all but S and $D \cap A = A'$. 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 DS on H-minor-free graphs [19, 31] outputs a set D of size at most $2(|S|+2)$ such that D dominates $V(G) \setminus (A \cup S)$ and $D \cap A = A'$. 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.

Lemma 13. *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 $G' \equiv_{\mathrm{DS},|S|} G$.*

Proof. Let the transposition constant be 0. To show that $G' \equiv_{\mathrm{DS},|S|} G$, we show that given a $|B|$-boundaried graph G_1 and a positive integer ℓ we have that $(G \oplus G_1, \ell) \in \mathrm{DS} \Leftrightarrow (G' \oplus G_1, \ell) \in \mathrm{DS}$. Let $Z \subset V(G \oplus G_1)$ be a dominating set for $G \oplus G_1$ of size at most ℓ. Let $Z_1 = V(G) \cap Z$. If $|Z_1| > |S|$ then $(Z \setminus Z_1) \cup S$ is a smaller dominating set for $G \oplus G_1$. Therefore we assume that $|Z_1| \leq |S|$. Let $A' = Z \cap A$, and observe that A' is feasible because Z_1 dominates all but S. If $w \notin Z$, then $Z' = Z$ is a dominating set of size at most ℓ for $G' \oplus G_1$. So assume $w \in Z$. Observe that $w \in Z_1$ and $w \notin S$ and therefore all the neighbors of w lie in G. 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 $N_{G \oplus G_1}(w) \setminus N_{G \oplus G_1}(Z \setminus w) \subseteq A$. There is a representative $w' \in R$, $w' \neq w$ (since $w \notin R$), such that $(N_{G \oplus G_1}(w) = N_G(w)) \cap A \subseteq N_G(w') \cap A$. Hence $Z' = (Z \cup \{w'\}) \setminus \{w\}$ is a dominating set of $G' \oplus G_1$ of size at most ℓ.

Now, let $Z' \subseteq V(G' \oplus G_1)$ be a dominating set of size at most ℓ for $G' \oplus G_1$. Let $Z'_1 = V(G') \cap Z'$. As in the forward direction we can assume that $|Z'_1| \leq |S|$. We show that Z' also dominates w in $G \oplus G_1$. Specifically $Z'_1 \cup \{w\}$ is a dominating set of all but S in G of size at most $|S| + 1$ so $Z'_1 \cap A$ is feasible. Since $\{w\}$ is irrelevant with respect to $Z'_1 \cap A$, we have $w \in N_G(Z'_1 \cap A)$ and thus Z' is a dominating set for $G' \oplus G_1$ of size at most ℓ. This concludes the proof. $\square$

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 13 and transitivity of $\equiv_{\mathrm{DS},t}$ we have that $G' \equiv_{\mathrm{DS},|S|} G$. We now prove that a graph G which can not be reduced by the irrelevant vertex rule has a property that each of its torso has a small 2-dominating set.

Lemma 14. *Let G be a graph which is irreducible by the Irrelevant Vertex Rule and S be a dominating set of G. For every torso $\tau(t)$ of the tree-decomposition (M, Ψ) of G, we have that $\tau(t) \setminus Z_t$ has a 2-dominating set of size $\mathcal{O}(|S|)$. Furthermore if G is a H-minor free graph then $\mathbf{tw}(G) = \mathcal{O}(\sqrt{|S|})$.*

Proof. Let $\tau(t)^* = \tau(t) \setminus A$, where A are the apices of $\tau(t)$. We will obtain a 2-dominating set of size $\mathcal{O}(|S|)$ in $\tau(t)^*$. 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 w. 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 $\tau(t)^*$ can be obtained from $G \setminus A$ by contracting all edges in $E(G \setminus A) \setminus E(\tau(t)^*)$ and adding all edges in $E(\tau(t)^*) \setminus E(G \setminus A)$. Since contracting and adding edges does not increase the size of a minimum 2-dominating set of a graph, $\tau(t)^*$ has a 2-dominating set of size $\mathcal{O}(|S|)$. This completes the proof for the first part.

Now assume that G is a H-minor free graph. It is well known that fact that the treewidth of a H-minor free graph is at most the maximum treewidth of its torsos, see e.g.[18]. Thus to show that $\mathbf{tw}(G) = \mathcal{O}(\sqrt{|S|})$ it is sufficient to show that its torsos has small treewidth. To conclude, $\tau(t)^*$ excludes an apex graph as a minor (see, e.g. [38, Theorem 13]) and it has a 2-dominating set of size $\mathcal{O}(|S|)$. By the bidimensionality of 2-dominating set, we have that $\mathbf{tw}(\tau(t)^*) = \mathcal{O}(\sqrt{|S|})$ [18, 29]. Now we add all the apices of A to all the bags of the tree decomposition of $\tau(t)^*$ to obtain a tree decomposition for $\tau(t)'$ of width $\mathcal{O}(\sqrt{|S|}) + h = \mathcal{O}(\sqrt{|S|})$. $\square$

Let us also remark that Irrelevant Vertex Rule is based on the performance of a polynomial time approximation algorithm. Thus by Lemmata 1, 13 and 14, and the fact that the treewidth of a graph is at most the maximum treewidth of its torsos, see e.g.[18], we obtain the following lemma.

Lemma 15. *There is a polynomial time algorithm that for a given graph G and a dominating set S of G, outputs graph G' such that $G' \equiv_{\mathrm{DS}} G$ and for every torso $\tau(t)$ of the tree-decomposition (M, Ψ) of G, we have that $\tau(t) \setminus Z_t$ has a 2-dominating set of size $\mathcal{O}(|S|)$. Furthermore if G is a H-minor free graph then $\mathbf{tw}(G) = \mathcal{O}(\sqrt{|S|})$.*

Having Lemma 15 proving Lemma 11 becomes simple.

Proof of Lemma 11. We apply Lemma 15 to G with a decomposition that has a single bag containing the entire graph and the apices A of the bag being the vertices of degree at least h'. By Lemma 15, $G \setminus A$ has a 2-dominating set of size $\delta_3|S|$. Since all vertices of $G \setminus A$ have degree at most h' it follows that $|V(G)| \leq h' + \delta_3 h|S|\delta_3 h^2|S| \leq \delta|S|$. $\square$

We need the following well known lemma, see e.g. [6], on separators in graphs of bounded treewidth for the proof of Lemma 12.

Lemma 16. *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 $\frac{w(V(G))-w(X)}{3} \leq w(V_i) \leq \frac{2(w(V(G))-w(X))}{3}$, for $i \in \{1,2\}$.*

Proof of Lemma 12. By (G, S) we denote the graph with boundary S. By Lemma 15, we may assume that $\mathbf{tw}(G) = \mathcal{O}(\sqrt{|S|})$. We prove the lemma using induction on $|S|$. If $|S| = \mathcal{O}(1)$ we are done, as in this case we know that G is a $|S|$-DS protrusion. Thus, if $|V(G)| > \xi_{|S|}$ then we can apply Lemma 5 and in polynomial time obtain a graph G^* such that $G^* \equiv_{\mathrm{DS}} G$ and $|V(G^*)| \leq \xi_{|S|}$. In the same time we can compute the translation constant depending on G and G^* and return it. Thus, we return G^* and the translation constant c.

Otherwise, using a constant factor approximation of treewidth on H-minor-free graphs [26], we compute a tree-decomposition of G of width $d\sqrt{|S|}$. Now, by applying Lemma 16 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, $|X| \leq d\sqrt{|S|} + 1$, and $|V_i \cap S| \leq 2|S|/3$ for $i \in \{1,2\}$. Let $S' = S \cup X$. Observe that S' is also a dominating set.

Let $S_1 = S' \cap (V_1 \cup X)$ and $S_2 = S' \cap (V_2 \cup X)$. Let $G_1 = G[V_1 \cup X]$ and $G_2 = G[V_2 \cup X]$. We now apply the algorithm recursively on (G_1, S_1) and (G_2, S_2) and obtain graphs G_1', G_2' such that for $i \in \{1,2\}$, $G_i \equiv_{\mathrm{DS},|S_i|} G_i$. Let c_1 and c_2 be the translation constants returned by the algorithm. Since $X \subseteq S'$, we have that S_i is a dominating set of G_i and hence we actually can run the algorithm recursively on the two subcases. The algorithm returns G_1' and G_2' and translation constants c_1 and c_2. Let $G' = G'_1 \oplus_\delta G'_2$ and $S' = S_1 \cup S_2$. We will show that $G' \equiv_{\mathrm{DS},|S'|} G$. Let G_3 be a graph with boundary S' and k be a positive integer. Then

$$\begin{aligned} ((G_1 \oplus_\delta G_2) \oplus G_3, k) &\in \mathrm{DS} \\ \iff ((G_1 \oplus_\delta G_3) \oplus G_2, k) &\in \mathrm{DS} \\ \iff ((G_1 \oplus_\delta G_3) \oplus G'_2, k + c_2) &\in \mathrm{DS} \\ \iff ((G'_2 \oplus_\delta G_3) \oplus G_1, k + c_2) &\in \mathrm{DS} \\ \iff ((G'_2 \oplus_\delta G_3) \oplus G'_1, k + c_2 + c_1) &\in \mathrm{DS} \\ \iff ((G'_2 \oplus_\delta G'_1) \oplus G_3, k + c_2 + c_1) &\in \mathrm{DS}. \end{aligned}$$

This proves that $G' \equiv_{\mathrm{DS},|S'|} G$. Now we will show that $|V(G')| \leq \mathcal{O}(|S|)$.

Let $\mu(|S|)$ be the largest possible size of the set $|V(G')|$ output by the algorithm when run on a graph G with a dominating set S. We upper bound $|V(G')|$ by the following recursive formula.

$$|V(G')| \leq \max_{1/3 \leq \alpha \leq 2/3} \left\{ \mu\left(\alpha|S| + d\sqrt{|S|}\right) + \mu\left((1-\alpha)|S|\right) + d\sqrt{|S|} \right\}.$$

Using simple induction one can show that the above solves to $\mathcal{O}(|S|)$. See for an example [31, Lemma 2]. Hence we conclude that $|V(G')| = \mathcal{O}(|S|) = \mathcal{O}(k)$. This completes the proof of the lemma. □

6.3 Kernelization algorithm for CDS

To obtain kernelization algorithm for CDS the only thing that remains to show are results analogous to Lemmas 12 and 11 for DS. However to obtain this we need to apply reduction rules developed in [34] for CDS. Finally we need to adapt the proofs of Lemmas 11, 12, 13 and 14 given in the full-version of [34] available at [35] with the new perspective. Two of these lemmas essentially shows the correctness of reduction rules for CDS and that every torsoes has 2-dominating set of size at most $\mathcal{O}(|S|)$. Here S is a connected dominating set of the input graph G. The only result that is not proved in [35] is the result analogous to Lemma 11 for DS. However, the size of a dominating set is at most the size of a connected dominating set. After this the proof for the case that given a graph G with at most h' vertices of degree at least h' we can return a canonically equivalent graph G' is verbatim to the proof of Lemma 11. We omit these adaptation details from this extended abstract.

Theorem 3. CDS *admits a linear kernel on graphs excluding a fixed graph H as a topological minor.*

7 Conclusions

In this paper we give linear kernels for two widely studied parameterized problems, namely DS and CDS, for every graph class that excludes some graph as a topological minor. The emerging questions are the following two:

1. Can our techniques be extended to more general sparse graph classes?
2. Can our techniques be applied to more general families of parameterized problems?

We believe that any step towards resolving the first question should be based on significant graph-theoretical advances. Our results make use of the decomposition theorem of Grohe and

Marx in [39] that, in turn, can be seen as an extension of seminal results of the Graph Minor Series by Robertson and Seymour [47]. So far no similar structural theorem is known for more general sparse graph classes. We also believe that a broadening of the kernelization horizon for these two problems without the use of some tree-based structural characterization of sparsity requires completely different ideas.

The first move towards resolving the second question is to extend our techniques for more variants of the dominating set problem. Natural candidates in this direction could be the r-Domination problem (asking for a set S that is within distance r from any other vertex of the graph), the Independent Domination problem (asking for a dominating set that induces an edgeless graph), or, more interestingly, the Cycle Domination problem (asking for a set S that dominates at least one vertex from each cycle of G). However, a more general meta-algorithmic framework, including general families of parameterized problems, seems to be far from reach.

Acknowledgement Thanks to Marek Cygan for sending us a copy of [13].

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] ——, *Linear time algorithms for finding a dominating set of fixed size in degenerated graphs*, Algorithmica, 54 (2009), pp. 544–556.

[5] S. Arnborg, B. Courcelle, A. Proskurowski, and D. Seese, *An algebraic theory of graph reduction*, Journal of the ACM, 40 (1993), pp. 1134–1164.

[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 and B. de Fluiter, *Reduction algorithms for constructing solutions in graphs with small treewidth*, 1996, pp. 199–208.

[8] 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.

[9] 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.

[10] H. L. Bodlaender and T. Hagerup, *Parallel algorithms with optimal speedup for bounded treewidth*, SIAM J. Comput., 27 (1998), pp. 1725–1746.

[11] H. L. Bodlaender and B. van Antwerpen-de Fluiter, *Reduction algorithms for graphs of small treewidth*, Information and Computation, 167 (2001), pp. 86–119.

[12] 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.

[13] M. Cygan, F. Grandoni, and D. Hermelin, *Tight kernel bounds for problems on graphs with small degeneracy*. Manuscript, 2012.

[14] 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.

[15] B. de Fluiter, *Algorithms for Graphs of Small Treewidth*, PhD thesis, Utrecht University, 1997.

[16] 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.

[17] 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.

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

[19] 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.

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

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

[22] 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.

[23] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Springer, 1998.

[24] A. Drucker, *New limits to classical and quantum instance compression*, in Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2012), IEEE, 2012, p. to appear.

[25] P. Duchet and H. Meyniel, *On Hadwiger's number and the stability number*, in Graph theory (Cambridge, 1981), vol. 62 of North-Holland Math. Stud., North-Holland, Amsterdam, 1982, pp. 71–73.

[26] U. Feige, M. Hajiaghayi, and J. R. Lee, *Improved approximation algorithms for minimum weight vertex separators*, SIAM J. Comput., 38 (2008), pp. 629–657.

[27] M. R. Fellows and M. A. Langston, *An analogue of the myhill-nerode theorem and its use in computing finite-basis characterizations (extended abstract)*, in FOCS, 1989, pp. 520–525.

[28] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.

[29] F. V. Fomin, P. A. Golovach, and D. M. Thilikos, *Contraction obstructions for treewidth*, J. Comb. Theory, Ser. B, 101 (2011), pp. 302–314.

[30] F. V. Fomin, D. Lokshtanov, N. Misra, G. Philip, and S. Saurabh, *Hitting forbidden minors: Approximation and kernelization*, in Proceedings of the 8th International Symposium on Theoretical Aspects of Computer Science (STACS 2011), vol. 9 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011, pp. 189–200.

[31] 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.

[32] F. V. Fomin, D. Lokshtanov, and S. Saurabh, *Bidimensionality and geometric graphs*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), SIAM, 2012, pp. 1563–1575.

[33] 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.

[34] ——, *Linear kernels for (connected) dominating set on H-minor-free graphs*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), ACM-SIAM, 2010, pp. 82–93.

[35] ——, *Linear kernels for (connected) dominating set on H-minor-free graphs*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), http://www.ii.uib.no/ daniello/.

[36] 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.

[37] P. A. Golovach and Y. Villanger, *Parameterized complexity for domination problems on degenerate graphs*, in WG, vol. 5344, 2008, pp. 195–205.

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

[39] M. Grohe and D. Marx, *Structure theorem and isomorphism test for graphs with excluded topological subgraphs*, in STOC, 2012, pp. 173–192.

[40] 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.

[41] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of domination in graphs*, Marcel Dekker Inc., New York, 1998.

[42] R. Impagliazzo, R. Paturi, and F. Zane, *Which problems have strongly exponential complexity?*, J. Comput. Syst. Sci., 63 (2001), pp. 512–530.

[43] M. Jones, D. Lokshtanov, M. S. Ramanujan, S. Saurabh, and O. Suchy, *Parameterized complexity of directed steiner tree on sparse graphs*, Manuscript, 2012.

[44] E. J. Kim, A. Langer, C. Paul, F. Reidl, P. Rossmanith, I. Sau, and S. Sikdar, *Linear kernels and single-exponential algorithms via protrusion decompositions*, CoRR, abs/1207.0835 (2012).

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

[46] 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.

[47] N. Robertson and P. D. Seymour, *Graph minors. XVI. Excluding a non-planar graph*, J. Combin. Theory Ser. B, 89 (2003), pp. 43–76.