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 apex 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. 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 and in fact prove a more general result. For Connected Dominating Set no polynomial kernel even on H-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];
- Our results are based on a generic reduction rule producing an equivalent instance (in case the input graph is H-minor-free) of the problem with treewidth $\mathcal{O}(\sqrt{k})$. The application of this rule in a divide-and-conquer fashion together with new notion of protrusions brings us to linear kernels.

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.

---

*Preliminary versions of this paper appeared in SODA 2012 and STACS 2013.

†University of Bergen, Norway, fomin@ii.uib.no. The research was supported by the European Research Council through ERC Grant Agreement n. 267959.

‡University of Bergen, Norway, daniello@ii.uib.no. The research was supported by the Bergen Research Foundation and the University of Bergen through project "BeHard'.

§The Institute of Mathematical Sciences, CIT Campus, Chennai, India, saket@imsc.res.in. The research was supported by the European Research Council through Starting Grant 306992 "Parameterized Approximation'.

¶Department of Mathematics, National and Kapodistrian University of Athens, Athens, Greece and AlGCo project-team, CNRS, LIRMM, Montpellier, France, sedthilk@thilikos.info. Co-financed by the E.U. (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: "Thales. Investing in knowledge society through the European Social Fund".

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 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 [11, 20, 31].

The Dominating Set (DS) problem together with its numerous variants, is one of the most classic and well-studied problems in algorithms and combinatorics [47]. 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. The connected variant of the problem, Connected Dominating Set (CDS) asks, given a graph G and a non-negative integer k, whether G contains a dominating set D of at most k vertices such that for every connected component C of G, we have that $G[V(C) \cap D]$ is connected. This definition of CDS differs slightly from the established one where one just demands that the subgraph induced by the dominating set be connected. Our definition generalizes the established one so to include disconnected graphs. 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 W[2]-complete and therefore it cannot be solved by a parameterized algorithm, unless an unexpected collapse occurs in the Parameterized Complexity hierarchy (see [28, 35, 54]) 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 [2, 21, 42] resulted in the creation of bidimensionality theory characterizing a broad range of graph problems that admit efficient approximation schemes, fixed-parameter algorithms or kernels on a broad range of graphs [22, 24, 27, 38, 40, 39].

One of the first results on linear kernels is the celebrated work of Alber et al. on DS on planar graphs [3]. This work augmented significantly the interest in proving polynomial (or preferably linear) kernels for other parameterized problems. The result of Alber et al. [3], see also [16], has been extended to a much more general graph classes like graphs of bounded genus [12] and apex-minor-free graphs [40]. An important step in this direction was done by Alon and Gutner [4, 46] 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 et al. [55] 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 et al. [55] produces a kernel of size $\mathcal{O}(k^{d^2})$. Similarly, the sizes of the kernels in [4, 46, 55] are bounded by polynomials in k with degrees depending on the size of the excluded minor H. Alon and Gutner [4] 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 [12], DS has a kernel of size $f(g) \cdot k$ on graphs of genus g. An alternative meta-algorithmic framework, based on bidimensionality theory [22], was introduced in [40], implying the existence of a kernel of size $f(H) \cdot k$ for DS on graphs excluding an apex[1] 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 and H-topological-minor-free graphs form much larger class than apex-minor-free graphs. For example, the class of graphs excluding $H = K_6$, the complete graph on 6 vertices, as a minor, contains all apex graphs. Alon and Gutner in [4] and Gutner in [46] posed as an open problem whether one can obtain a linear kernel for DS on H-minor-free graphs. 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 [4] for the special case where $H = K_{3,h}$. See Fig. 1 for the relationship between these classes.

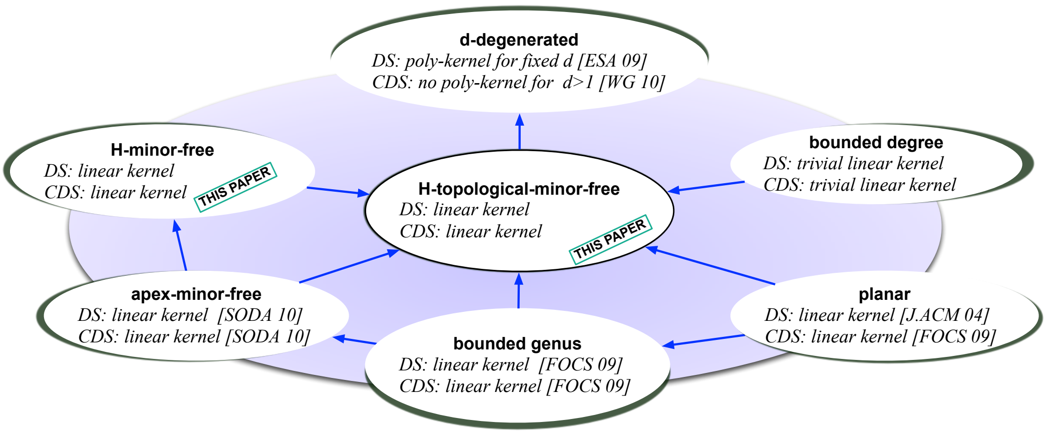

Figure 1: Kernels for DS and CDS on classes of sparse graphs. Arrows represent inclusions of classes. In the diagram, [J.ACM 04] is referred to the paper of Albers et al. [3], [FOCS 09] to the paper of Bodlaender et al. [12], [SODA 10] and [SODA 12] to the papers of Fomin et al. [40] and [41], [ESA 09] to the paper of Philip et al. [55], and [WG 10] to Cygan et al. [18].

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 recent work of Cygan et al. [17], the kernelization algorithm of Philip et al. [55] is essentially tight—existence of a kernel of size $\mathcal{O}(k^{(d-3)(d-1)-\varepsilon})$, would imply that coNP is in $\mathsf{NP/poly}$. In spite of these negative news, we show how to lift the linearity of kernelization for DS from bounded-degree graphs and apex 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 $\mathsf{NP/poly}$ [18].

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

[1] An *apex* graph is a graph that can become planar if we remove some of its vertices.

DS on this class of graphs significantly extends and improves previous works [4, 41, 46]. The extension of the results for planar graphs from [3] and apex-minor-free graphs from [40] 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 [25] and parameterized algorithms [22, 29]. The basic idea behind kernelization algorithms on apex-minor-free graphs is the bidimensionality of DS. Roughly speaking, the treewidth of these graphs with dominating set k is $o(k)$. In other words, 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.

One of the main ingredient of our kernelization algorithms is new reduction rules that allow us to obtain the desired kernels on 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 whose 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 [40] 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 [3, 16] 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 [56]. 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 [50, 51, 53]. Moreover, Adler et al. [1] 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 [57] 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 kernel. To overcome this obstacle, we have to implement the irrelevant vertex rule in a divide and conquer manner, and only after doing this we can guarantee that the reduced graph admits a linear kernel. The idea of using divide and conquer in kernelization is our first conceptual contribution.

The ideas introduced for H-minor-free graphs 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 [48]. 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 second 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 in [12, 40, 37, 52]. At this point it is also important to mention earlier works done in [34, 7, 15, 19, 14] 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 explained above. 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 generalized notion of protrusion, which is the third conceptual contribution of this paper. In the new notion of protrusion we relax the requirement that protrusions are of bounded treewidth by the condition that they have a bounded dominating set. Let us remark, that a similar notion of a generalized 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 generalized protrusion, which can be replaced by a smaller equivalent subgraph. The construction of the partitioning is heavily based on the recent work of Grohe and Marx on the structure of such graphs [45].

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 $2^{\mathcal{O}(\sqrt{k})} + n^{\mathcal{O}(1)}$. 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. [22]. Also our kernels can be used to obtain subexponential parameterized algorithms for these problems that use polynomial space.

Organization of the paper. The remaining part of this paper is organized as follows. In Section 2, we provide definitions and state known results used in the paper. In Section 3, we introduce the new notion of "generalized protrusions" and build a theory of replacements for such protrusions. We provide a decomposition lemma in Section 4, which will be used to for kernelization algorithms. In Sections 5 and 6 we give the two main results of the paper, linear kernels for DS and CDS. In Section 7 we conclude with questions for further research and give a short overview of developments occurred since the conference versions of this paper were published, including work on kernelization of DS and CDS on graphs of bounded expansion and nowhere-dense graphs.

2 Preliminaries

In this section we give various definitions which we make use of in the paper. We refer to Diestel's book [26] 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 a 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. A subset of vertices D is called a *dominating set* of G if $N[D] = V(G)$. A subset of vertices D is called *connected dominating set* if it is a dominating set and for every connected component C of G we have that $G[D \cap C]$ is connected. 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 a 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.

Throughout this paper we use $\mathbb{Z}$, $\mathbb{Z}^+$ and $\mathbb{Z}^-$ for the sets of integers, non-negative and non-positive integers respectively. Finally, we use $\mathbb{N}$ for the set of positive integers.

Minors and Contractions. 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 is a pair (M, Ψ) where M is a rooted tree and $\Psi : V(M) \rightarrow 2^{V(G)}$, such that :

1. $\bigcup_{t \in V(M)} \Psi(t) = V(G)$.
2. For each edge $uv \in E(G)$, there is a $t \in V(M)$ such that both u and v belong to $\Psi(t)$.
3. For each $v \in V(G)$, 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 [45]. Given a tree decomposition of graph G, we define mappings $\sigma, \gamma : V(M) \rightarrow 2^{V(G)}$ and $\kappa : E(M) \rightarrow 2^{V(G)}$. 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 $e = uv \in E(M)$, $\kappa(e) = \Psi(u) \cap \Psi(v)$.
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

$$\min \Big\{ |\Psi(t)| - 1 \mid t \in V(M) \Big\},$$

and the *adhesion* of the tree decomposition is

$$\max \Big\{ |\sigma(t)| \mid t \in V(M) \Big\}.$$

We use $\mathbf{tw}(G)$ to denote the treewidth of the input graph. For every node $t \in V(M)$, the *torso* at t is the graph

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

Parameterized graph problems. A parameterized graph problem Π in general 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). In this paper we use an extension of this definition used in [13] that permits the parameter k to be negative with the additional constraint that either all pairs with non-positive value of the parameter are in Π or that no such pair is in Π. Formally, a parametrized problem Π is a subset of $\Sigma^* \times \mathbb{Z}$ where for all $(x_1, k_1), (x_2, k_2) \in \Sigma^* \times \mathbb{Z}$ with $k_1, k_2 < 0$ it holds that $(x_1, k_1) \in \Pi$ if and only if $(x_2, k_2) \in \Pi$. This extended definition encompasses the traditional one and is being adopted for technical reasons (see Subsection 3.2). In an instance of a parameterized problem (x, k), the integer k is called the parameter.

Kernels and Protrusions. A central notion in parameterized complexity is *fixed parameter tractability*, 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. The notion of *kernelization* is formally defined as follows.

Definition 1. *A* kernelization algorithm*, or simply a* kernel*, for a parameterized problem Π is an algorithm $\mathcal{A}$ that, given an instance (x, k) of Π, works in polynomial-time and returns an equivalent instance (x', k') of Π. Moreover, there exists a computable function $g(\cdot)$ such that whenever (x', k') is the output for an instance (x, k), then it holds that $|x'| + k' \leq g(k)$. If the upper bound $g(\cdot)$ is a polynomial (linear) function of the parameter, then we say that Π admits a* polynomial (linear) kernel*.*

We often abuse the notation and call the output of a kernelization algorithm, the "reduced" equivalent instance, also a kernel.

Definition 2. *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.*

2.1 Known Decomposition Theorems

We start by the definition of nearly embeddable graphs.

Definition 3 (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 ([45, 57])**.** *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-free graph 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 $f(|H|)n^{\mathcal{O}(1)}$ 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.

One of the main consequence of Theorem 1 we need for our purposes is that (in the case when G is H-minor-free) 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. [44, Theorem 13].

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 [35, Lemma 11.9] for the proof. Finally, we also need the following result.

Lemma 1 ([57])**.** *For any two graphs G_1 and G_2, $\mathbf{tw}(G_1 \oplus G_2) \leq \max\{\mathbf{tw}(G_1), \mathbf{tw}(G_2)\}$.*

2.2 Known Approximation Algorithms

Recall that by $\mathcal{C}_H$ we denote the class of graphs that exclude H as a topological subgraph. In this subsection we mention a constant factor approximation for DS on $\mathcal{C}_H$. It is well known that graphs in $\mathcal{C}_H$ has bounded degeneracy. The following is know about the approximation of DS.

Lemma 2 ([49])**.** *Let H be a graph. Then there exists a constant $\eta(H)$ depending only on $|H|$ such that* Dominating Set*, admits a $\eta(H)$-factor approximation algorithm on $\mathcal{C}_H$.*

For CDS we need the following proposition attributed to [32].

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 2 and Proposition 1 we arrive to the following.

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

3 Generalized Protrusions

In this section we introduce a notion of "generalized protrusion" and build a theory of replacement for them. We first start with some relevant definitions and concepts.

3.1 Boundaried Graphs

Here we define the notion of *boundaried graphs* and various operations on them.

Definition 4. [**Boundaried Graphs**] *A boundaried graph is a graph G with a set $B \subseteq V(G)$ of distinguished vertices and an injective labelling λ from B to the set $\mathbb{Z}^+$. The set B is called the* boundary *of G and the vertices in B are called* boundary vertices *or* terminals. *Given a boundaried graph G, we denote its boundary by $\delta(G)$, we denote its labelling by λ_G, and we define its* label set *by $\Lambda(G) = \{\lambda_G(v) \mid v \in \delta(G)\}$. Given a finite set $I \subseteq \mathbb{Z}^+$, we define $\mathcal{F}_I$ to denote the class of all boundaried graphs whose label set is I. Similarly, we define $\mathcal{F}_{\subseteq I} = \bigcup_{I' \subseteq I} \mathcal{F}_{I'}$. We also denote by $\mathcal{F}$ the class of all boundaried graphs. Finally we say that a boundaried graph is a t-*boundaried *graph if $\Lambda(G) \subseteq \{1, \dots, t\}$.*

Definition 5. [**Gluing by $\oplus$**] *Let G_1 and G_2 be two boundaried graphs. We denote by $G_1 \oplus G_2$ the graph (not boundaried) 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 either an edge between them in G_1 or in G_2.*

Definition 6. [**Gluing by $\oplus_\delta$**] *The* boundaried gluing operation *$\oplus_\delta$ is similar to the normal gluing operation, but results in a boundaried graph rather than a graph. Specifically $G_1 \oplus_\delta G_2$ results in a 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 to a graph class $\mathcal{G}$* if the underlying graph belongs to $\mathcal{G}$.

Definition 1. [**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 $\delta(G^\star) = \delta(G)$.*

3.2 Finite Integer Index

Definition 7. [**Canonical equivalence on boundaried graphs.**] *Let Π be a parameterized graph problem whose instances are pairs of the form (G, k). Given two boundaried graphs $G_1, G_2 \in \mathcal{F}$, we say that $G_1 \equiv_\Pi G_2$ if $\Lambda(G_1) = \Lambda(G_2)$ and there exists a* transposition constant *$c \in \mathbb{Z}$ such that*

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

Note that the relation $\equiv_\Pi$ is an equivalence relation. Observe that c could be negative in the above definition. This is the reason we gave the definition of parameterized problems to include negative parameters also.

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

Definition 8. [**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* $\equiv_\Pi$. *We say that* $J \in \mathcal{C}$ *is a* progressive representative *of* $\mathcal{C}$ *if for every* $H \in \mathcal{C}$ *there exists* $c \in \mathbb{Z}^-$, *such that*

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

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

Lemma 4 ([12])**.** *Let* Π *be a parameterized graph problem whose instances are pairs of the form* (G, k). *Then each equivalence class of* $\equiv_\Pi$ *has a progressive representative.*

Notice that two boundaried graphs with different label sets belong to different equivalence classes of $\equiv_\Pi$. Hence for every equivalence class $\mathcal{C}$ of $\equiv_\Pi$ there exists some finite set $I \subseteq \mathbb{Z}^+$ such that $\mathcal{C} \subseteq \mathcal{F}_I$. We are now in position to give the following definition.

Definition 9. [**Finite Integer Index**] *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 finite* $I \subseteq \mathbb{Z}^+$, *the number of equivalence classes of* $\equiv_\Pi$ *that are subsets of* $\mathcal{F}_I$ *is finite. For each* $I \subseteq \mathbb{Z}^+$, *we define* $\mathcal{S}_I$ *to be a set containing exactly one progressive representative of each equivalence class of* $\equiv_\Pi$ *that is a subset of* $\mathcal{F}_I$. *We also define* $\mathcal{S}_{\subseteq I} = \bigcup_{I' \subseteq I} \mathcal{S}_{I'}$.

3.3 Replacement lemma

We first define a notion of monotonicity for parameterized problems.

Definition 2. *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 3. *Let* Π *be a monotone parameterized graph problem that is FII. Let* $\mathcal{S}_t$ *be a set containing exactly one progressive representative of each equivalence class of* $\equiv_\Pi$ *that is a subset of* $\mathcal{F}_I$, *where* $I = \{1, \dots, t\}$. *For a* t*-boundaried graph* G, *we define*

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

The main result of this section is as follows.

Lemma 5. *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* $\mathcal{O}(\iota(G)(f(|V(G)| + \xi_t, \iota(G)))$ *steps, a* t*-boundaried graph* G^* *such that* $G \equiv_\Pi G^*$ *and* $|V(G^*)| < \xi_t$. *Moreover we can compute the translation constant* c *from* G *to* G^* *in the same time.*

Proof. We give proof only for positive monotone problem Π, the proof for negative monotone is identical. Let $\mathcal{S}_t$ be a set containing exactly one progressive representative of each equivalence class of $\equiv_\Pi$ that is a subset of $\mathcal{F}_I$, where $I = \{1, \ldots, t\}$ and 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

$$\forall (F, k) \in \mathcal{F}_t \times \mathbb{Z} \qquad (G \oplus F, k) \in \Pi \Leftrightarrow (Y_\ell \oplus F, k - \eta(X, Y_\ell)) \in \Pi. \tag{2}$$

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 $k \leq \iota(G)$. Thus the total time to compute the $\mathcal{X}$ is $\mathcal{O}(\iota(G)(f(|V(G)| + \xi_t, \iota(G)))$.

- 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 $G \equiv_\Pi G^*$. This completes the proof. $\square$

We remark that the algorithm whose existence is guaranteed by the Lemma 5 assumes that the set $\mathcal{S}_t$ of representatives are hardwired in the algorithm and that in general there is no procedure that for problems having FII outputs such a representative set.

4 Slice-Decomposition

In this section our objective is to show that in polynomial time we can partition the graph G into parts that satisfy 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 4. [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 by X is small. We can similarly define the notion of r-Π-protrusion for various other graph problems Π.

Definition 5. [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 connected and is a 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. We will also need the following. Let $\mathcal{G}$ be a graph class. Given a parameterized graph problem Π and a graph class $\mathcal{G}$, we denote by $\Pi \Cap \mathcal{G}$ the problem obtained by removing from Π all instances that encode graphs that do not belong to $\mathcal{G}$. Our result is as follows.

Lemma 6. *Let H be a fixed graph. For every $t \in \mathbb{Z}^+$, 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 $H \not\preceq_T X'$ and $X \equiv_{\mathrm{DS}} X'$ ($X \equiv_{\mathrm{CDS}} X'$) and $|X'| \leq \xi_t$. Moreover in the same time we can also find the translation constant c from X to X'.*

Proof. Let $\mathcal{G}$ be the class of graphs that excludes H as a topological minor (or just as a minor). For every $t \in \mathbb{Z}^+$ let ξ_t be the constant as defined in Lemma 5. It is also known that both DS$\Cap\mathcal{G}$ (CDS$\Cap\mathcal{G}$) are FII [12, 13] and monotone. Furthermore, DS and CDS can be solved in time $\mathcal{O}((hk)^{hk}n)$ [5, Theorem 4] and $\mathcal{O}(k^{\mathcal{O}(h^2)k}n^{\mathcal{O}(1)})$ [43, Theorem 1] respectively. Thus, if $|X| > \xi_t$ then by Lemma 5 in time $\mathcal{O}(|X|)$ ($|X|^{\mathcal{O}(1)}$)), we can obtain a t-boundaried graph X' such that $X \equiv_{\mathrm{DS}} X'$ ($X \equiv_{\mathrm{CDS}} X'$) 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 5. □

We would like to remark here that in Lemma 6 if X excluded a fixed graph H as a minor then we could find X' such that $X \equiv_{\mathrm{DS}} X'$ and X' excludes H as a minor.

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

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

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

Definition 6. [(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 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.*
- $$\sum_{i=1}^{\alpha} \left(\sum_{e \in \mathcal{E}(M_i)} |\kappa(e)| \right) \leq \beta.$$

We refer to sets R_i^+, $i \in \{1, \ldots, \alpha\}$, as to 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.

1. Apply Lemma 2 (Lemma 3) 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.

Definition 7. *Let (M, Ψ) be the tree decomposition of a graph G given by 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 6 or a similar lemma developed in [12, 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 7. *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 contradicts 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

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

follows from Fact 1. □

Lemma 8. *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 2 (Lemma 3).*

Proof. Root the tree at an arbitrary node r of degree at least 3. 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 with $w_1, \ldots, w_\ell$, 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 9. *Let G be a graph such that $H \not\preceq_T G$ and (M, Ψ) be the tree decomposition given by Theorem 1. If M' is a subtree of M, then*

$$(D \cap \Psi(M')) \bigcup_{e \in \mathcal{E}(M')} \kappa(e)$$

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,

$$(D \cap \Psi(M')) \bigcup_{e \in \mathcal{E}(M')} \kappa(e)$$

is a dominating set for $G[\Psi(M')]$. □

Let $P_1, \ldots, P_\ell$ 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 10. *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. Targeting towards a contradiction we assume that for some $i \in \{1,\ldots,\ell\}$, we have $|P_i| > 2\xi_{2h}(|D \cap \Psi(M(P_i))|$. Let $P_i := s_i = a^i_1 \cdots a^i_{t_i} = t_i$. For every vertex

$$w \in \Big(D \cap \Psi(M(P_i))\Big) \bigcup \kappa(s_i s_i^*) \bigcup \kappa(t_i t_i^*)$$

we mark two vertices of the path P_i. We mark the first and the last vertices on P_i, say a^i_{wfirst} and a^i_{wlast}, such that $w \in \Psi(a^i_{\mathsf{wfirst}})$ and $w \in \Psi(a^i_{\mathsf{wlast}})$. That is, $w \in \Psi(a^i_{\mathsf{wfirst}})$ and $w \in \Psi(a^i_{\mathsf{wlast}})$ and for all $z < \mathsf{wfirst}$ or $z > \mathsf{wlast}$ we have that $w \notin \Psi(a^i_z)$. 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 the pigeonhole principle we have that there exists a subpath of P_i, say $P_i(a^i_x, a^i_y)$, such that no vertex of this subpath is marked and $|P_i(a^i_x, a^i_y)| > \xi_{2h}$. Let $W = \Psi(M(P_i(a^i_x, a^i_y)))$. Let a^* and b^* be the neighbors of a^i_x and a^i_y respectively that are not present on $P_i(a^i_x, a^i_y)$. Clearly, the only vertices in W that have neighbors in $V(G) \setminus W$ belong to $\kappa(a^i_x a^*) \cup \kappa(a^i_y b^*)$. Thus it is upper bounded by $2h$. Furthermore, since no vertex on the path $P_i(a^i_x, a^i_y)$ is marked, we have that all the vertices in D belonging to W also belongs to $\kappa(a^i_x a^*) \cup \kappa(a^i_y b^*)$. Then by Lemma 9, we have that $\kappa(a^i_x a^*) \cup \kappa(a^i_y b^*)$ 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 the desired size. □

The final result of this section is the following decomposition lemma.

Lemma 11. *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*

- *a $(\delta_1 k, \delta_2 k)$-slice decomposition; or*
- *a $2h$-*DS*-protrusion (or $2h$-*CDS*-protrusion) of size more than ξ_{2h} or;*
- *an 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 2 (Lemma 3) is at most $\eta(H)k$. Now we apply the marking scheme as described in Figure 2. Let M^* be the subtree of M formed by heavy edges. By Lemma 8, we know that

(a) $|L(M^*)| \le \eta(H)k$;

(b) $|S_{\ge 3}(M^*)| \le \eta(H)k - 1$; and

(c) $|\mathscr{P}_2(M^*)| \le 2\eta(H)k - 1$.

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 10 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 $|P_i| \le \xi_{2h}2(2h + k_i)$.

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 $i \ne j$. Furthermore, for all $i \ne j$ we have that

$$\Big|\Psi(M(P_i)) \cap D \cap \Psi(M(P_j)\Big| \le h.$$

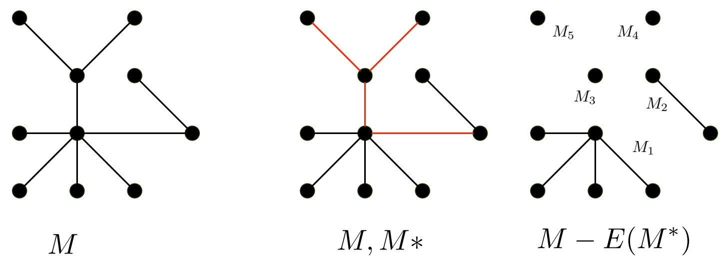

Figure 3: An illustration of the decomposition. The heavy edges are shown in red.

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 $\mathcal{F}$, see Fig. 3 for an illustration. Note that

$$\alpha \leq (2 + 12h\xi_{2h} + 2\xi_{2h})\eta(H)k.$$

We now argue that either the collection $M_1, \ldots, M_\alpha$ forms a $(\delta_1 k, \delta_2 k)$-slice decomposition of G or we found a $2h$-protrusion or a $2h$-DS-protrusion of size more than ξ_{2h} in G.

First we show that

$$\sum_{i=1}^{\alpha} \Big(\sum_{e \in \mathcal{E}(M_i)} |\kappa(e)| \Big) = \mathcal{O}(k).$$

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 incident with 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 9, we have that $(D \cap W) \cup \kappa(r_i w)$ 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 $\kappa(r_i w)$ 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 that there exists an h^* depending only on h such that either $\tau(M_i)$ has at most $h^* = \xi_{h+\xi_{2h}} + h$ vertices of degree larger than h^*, or G contains an h'-protrusion of size more than $\xi_{h'}$. Here, $h' = \xi_{2h} + h$. 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 $|N' \setminus N| \geq \xi_{h+\xi_{2h}}$ and hence $|X| \geq \xi_{h+\xi_{2h}}$. 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. □

5 Kernelization Algorithm for DS

In this section we use slice-decomposition obtained in the last section to obtain linear kernels for DS and in the next section outline an algorithm for CDS.

Given an instance (G, k) of DS we first apply Lemma 2 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 11 and

- either find a $(\delta_1 k, \delta_2 k)$-slice decomposition; or
- a $2h$-DS-protrusion X of G 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 6. Given X, by making use of Lemma 6, we obtain a boundaried graph X' such that $|X'| \leq \xi_{2h}$ and $X \equiv_{\mathrm{DS}} X'$. 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 [12, 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 11 returns a slice-decomposition. For simplicity we denote by (G, k) itself the graph on which Lemma 11 returns the slice-decomposition. Since the number of times this process can be repeated is upper bounded by $n = |V(G)|$, we can obtain a $(\delta_1 k, \delta_2 k)$-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 9, 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 12. *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*

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

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

The second lemma is for H-minor-free graphs.

Lemma 13. *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*

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

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

Once we have proved Lemmata 12 and 13, we construct 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 12 or Lemma 13 based on the type of G_i and obtain a graph G'_i such that $G'_i \equiv_{\text{DS}} G_i$ 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 13 or 12 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 Lemmata 12 and 13 to complete the proof of Theorem 2.

5.1 Irrelevant Vertex Rule and proofs for Lemmas 12 and 13

For the proofs of Lemmas 12 and 13 we will introduce a reduction rule that removes irrelevant vertices. 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 6 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 [23, 38] 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 14. *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 $G' \equiv_{\mathrm{DS}} G$.*

Proof. We view G and G' as graphs with boundary S. Let the transposition constant be 0. To prove that $G' \equiv_{\mathrm{DS}} G$, 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. ☐

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 14 and transitivity of $\equiv_{\mathrm{DS}}$ we have that $G' \equiv_{\mathrm{DS}} G$. 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 15. *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 the treewidth of a H-minor-free graph is at most the maximum treewidth of its torsos, see e.g.[22]. Thus to show that $\mathbf{tw}(G) = \mathcal{O}(\sqrt{|S|})$ it is sufficient to show that its torsos have small treewidth. To conclude, $\tau(t)^*$ excludes an apex graph as a minor (see, e.g. [44, Theorem 13]) and it has a 2-dominating set of size $\mathcal{O}(|S|)$. By the bidimensionality of 2-dominating set, we have that $\mathbf{tw}(\tau(t)^*) = \mathcal{O}(\sqrt{|S|})$ [22, 36]. 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 $\mathcal{O}(\sqrt{|S|}) + h = \mathcal{O}(\sqrt{|S|})$. □

Let us also remark that Irrelevant Vertex Rule is based on the performance of a polynomial time approximation algorithm. Thus by Lemmata 2, 14 and 15, and the fact that the treewidth of a graph is at most the maximum treewidth of its torsos, see e.g.[22], we obtain the following lemma.

Lemma 16. *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 $\mathbf{tw}(G) = \mathcal{O}(\sqrt{|S|})$.*

Before we proceed further, we show the power of Lemma 16 by deriving a simple subexponential time algorithm for DS on H-minor-free graphs. This is one of the cornerstone results in [22] and is based on a non-trivial two-layer dynamic programming over clique-sum decomposition tree of a H-minor-free graphs. Lemma 16 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 [23, 38] 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 16 and obtain an equivalent graph G' such that $\mathbf{tw}(G') = \mathcal{O}(\sqrt{k})$. Now applying a constant factor approximation algorithm developed in [22] 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 $\mathcal{O}(3^t n^{\mathcal{O}(1)})$ [58]. This together with the above bound on the treewidth, gives us an alternative proof of the following theorem.

Theorem 3 ([23]). *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 $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$.*

Having Lemma 16 proving Lemma 12 becomes simple.

Proof of Lemma 12. We apply Lemma 16 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 16, $G \setminus A$ 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 $|V(G)| \leq h' + \delta_3 h|S|\delta_3 h^2|S| \leq \delta|S|$. □

We need the following well known lemma, see e.g. [9], on separators in graphs of bounded treewidth for the proof of Lemma 13.

Lemma 17. *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 13. By (G,S) we denote the graph with boundary S. By Lemma 16, 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 6 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 [33], we compute a tree-decomposition of G of width $d\sqrt{|S|}$. Now, by applying Lemma 17 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 $i \in \{1,2\}$, $G_i \equiv_{\mathrm{DS}} G_i$. 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 $G' \equiv_{\mathrm{DS}} G$. 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 $G' \equiv_{\mathrm{DS}} G$. 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 [38, Lemma 2]. Hence we conclude that $|V(G')| = \mathcal{O}(|S|) = \mathcal{O}(k)$. This completes the proof of the lemma. □

The algorithm of Demaine et al. [23] 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 2 implies almost directly the following refinement of Theorem 3.

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 2 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. [6] 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.

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

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

6 Kernelization algorithm for CDS

The kernelization algorithm for CDS is similar to DS—we also use slice-decomposition to obtain a linear kernel. However, the irrelevant vertex rule is a bit different. The kernelization algorithm for CDS follows from the results analogous to Lemmata 13 and 12 for DS. For completeness we spell out all the steps.

In particular given an instance (G, k) of CDS we first apply Lemma 2 and find a dominating set D of G. If $|D| > \eta(H)k$ we return that (G, k) is a NO instance to CDS. Else, we apply Lemma 11 and

- either find $(\delta_1 k, \delta_2 k)$-slice decomposition; or
- a $2h$-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 6. For a given X, we apply Lemma 6 and construct a boundaried graph X' such that $|X'| \leq \xi_{2h}$ and $X \equiv_{\text{CDS}} X'$. 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 $(G', k + c)$, here we remind that C is a negative integer. In the third case we apply the protrusion replacement lemma of [12, 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 11 returns a slice-decomposition. For simplicity we denote by (G, k) itself the graph on which Lemma 11 returns the slice-decomposition. The number of times this process can be repeated does not exceed $n = |V(G)|$ and a $(\delta_1 k, \delta_2 k)$-slice decomposition for (G, k) is constructed in polynomial time.

The pairwise disjoint connected subtrees $\{M_1, \ldots, M_\alpha\}$ of M coming from the slice-decomposition of G is denoted by $\mathcal{P}$ and we put $R_i^+ = \Psi(M_i)$. We define $Q_i = \bigcup_{e \in \mathcal{E}(M_i)} \kappa(e)$, $B_i = (D \cap R_i^+) \cup Q_i$ and $b_i = |B_i|$. As in the previous section, we treat $G_i := G[R_i^+]$ as a graph with boundary B_i. Then by Lemma 9, B_i is a dominating set for G_i.

For two kinds of graphs G_i, we use different reductions. In the first case we have that the graph G_i has at most h' vertices of degree at least h'.

Lemma 18. *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*

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

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

In the other case we have that G_i is H^*-minor-free for a graph H^* whose size only depends on h.

Lemma 19. *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*

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

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

In order to obtain the linear sized kernel for CDS the proof of Lemmata 18 and 19 sufficies. Indeed, for 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 18 or Lemma 19 based on the type of G_i and obtain a graph G_i' such that $G_i' \equiv_{\mathrm{CDS}} G_i$ and $|V(G_i')| \leq \delta b_i$. We view $G = G_i \oplus G^\star$, 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 19 or 18 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

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

Thus (subject to the proof of two lemmata) we have the following theorem.

Theorem 5. CDS *admits a linear kernel on graphs excluding a fixed graph H as a topological minor.*

6.1 Irrelevant Vertex Rule and proofs for Lemmata 18 and 19

As with DS, we will reduce the treewidth of a torso not only in the beginning of the procedure but also when we apply it recursively. 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 14, 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 [8]. Alternatively we can test it in time $n^{4h+\mathcal{O}(1)}$ by brute force. Let S_{4h} denote the set of vertices in $N^{4h}_{G\setminus A}[S] = N^{4h}_{G\setminus A}[S \setminus A]$. The set of vertices *covered* by A' is $W(A') = N[A'] \setminus (A \cup S \cup S_{4h})$. Note that a vertex in $N^{4h}_{G\setminus A}[S]$ is never covered by a set A'. Let $\mathsf{CutVert}$ denote the set of vertices w in G such that $G - \{w\}$ has more connected components than G. Observe that if G will be connected then $\mathsf{CutVert}$ is essentially the set of cut vertices. However, for disconnected graph it is the union of cut vertices for each connected component.

The definition of an irrelevant vertex with respect to A is different than for DS. A vertex

$$w \notin (S \cup S_{4h} \cup R \cup \mathsf{CutVert})$$

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 5 for DS but the correctness 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.

Lemma 20. *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 $G' \equiv_{\mathrm{CDS}} G$.*

Proof. We view G and G' as graphs with boundary S. Let the transposition constant be 0. To show that $G' \equiv_{\mathrm{CDS}} G$, we show that given any boundaried graph G_1 and a positive integer ℓ we have that $(G \oplus G_1, \ell) \in \mathrm{CDS} \Leftrightarrow (G' \oplus G_1, \ell) \in \mathrm{CDS}$. Let $Z \subset V(G \oplus G_1)$ be a connected dominating set for $G \oplus G_1$ of size at most ℓ. Observe that since S is a dominating set of G, we have that there exists a connected dominating set $S \subseteq S^*$ such that $|S^*| \leq 3|S|$ (Proposition 1). Let $Z_1 = V(G) \cap Z$. If $|Z_1| > 3|S|$ then $(Z \setminus Z_1) \cup S^*$ is a smaller connected dominating set for $G \oplus G_1$. Thus, we assume that $|Z_1| \leq 3|S|$. Let $A' = Z_1 \cap A$, and observe that A' is feasible since Z_1 dominates all but S and has size at most $3|S|$. If $w \notin Z$, then $Z' = Z$ is a connected dominating set of size ℓ for $G' \oplus G_1$. 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 Q be the connected component of $G \oplus G_1$ that contains w. Since, w is not a cut vertex of G, we have the following easy observation.

Observation 1. *$Q \setminus \{w\}$ is connected.*

Let $Z_Q = Z \cap Q$ be the connected dominating set of Q, $|Z_Q| = p$. We will show that $Q \setminus \{w\}$ has a connected dominating set of size at most p and that will show that $(G' \oplus G_1, \ell) \in \mathrm{CDS}$.

Observe that since $w \in W(A')$ and the only vertices that are common between G and G_1 belong to S, we have that $N^{4h}_{(G \oplus G_1)\setminus A}[w] = N^{4h}_{G \setminus A}[w] \subseteq V(G) \setminus S_{3h}$.

Let X be the vertex set of the connected component of $G \oplus G_1[Z_Q \cap N^{4h}_{G\setminus A}[w]]$ 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 $N_G(X') \cap A \supseteq N_G(X) \cap A$. Furthermore, since $|X| < 4h$ we have that every connected component of $G \oplus G_1[Z_Q \setminus X]$ contains a vertex of A'. This implies that $Z'_Q = (Z_Q \setminus X) \cup X'$ is connected. Since $X \subseteq W(A')$, and $|X| < 4h$ we have that $N_{G \oplus G_1}(X) = N_G(X)$. This implies that $N_G(X) \subseteq N_G(X' \cup A') \subseteq N_{G \oplus G_1}(X' \cup A')$ and thus Z'_Q is a connected dominating set of size at most p of Q that avoids w and thus by Observation 1, it is also a connected dominating set of $Q \setminus \{w\}$. This implies that in this case $(G' \oplus G_1, \ell) \in$ CDS.

Now suppose that $|X| \geq 4h$. Let $A^* = N_G(X) \cap A$. The vertex set A^* is a dominating set of size at most h in the connected 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 $|X \setminus Y| \geq 4h - 1$ (as every path from w to a vertex in Y has length at least $4h - 1$) and that $N_G[Y] \cap A \subseteq A^*$. Set $X' = (X^* \setminus P) \cup P'$, and $Z'_Q = (Z_Q \setminus (X \setminus Y)) \cup X'$. We have that $|X'| \leq |X^*| \leq 3h$ while $|X \setminus Y| \geq 4h - 1 \geq 3h$. Hence $|Z'_Q| \leq |Z_Q|$. Note that $G[X']$ is connected. Furthermore by our choice of $(X \setminus Y)$ we have that every connected component of $G \oplus G_1[Z_Q \setminus X]$ contains a vertex of Y and hence a vertex of A^*. However, $A^* \subseteq X'$ and $G[X']$ (or $G \oplus G_1[X']$) is connected and thus $G \oplus G_1[Z'_Q]$ is connected. Observe that $N_{G \oplus G_1}(X \setminus Y) = N_G(X \setminus Y)$. This implies that $N_G(X \setminus Y) \subseteq N_G(X' \cup A^*) \subseteq N_{G \oplus G_1}(X' \cup A^*)$ and thus Z'_Q is a connected dominating set of size at most p of Q that avoids w and thus by Observation 1 is also a connected dominating set of $Q \setminus \{w\}$. This implies that in this case $(G' \oplus G_1, \ell) \in$ CDS.

Now we prove the reverse direction. Let $Z' \subset V(G' \oplus G_1)$ be a connected dominating set for $G' \oplus G_1$ of size at most ℓ. By Observation 1 we know that $Q \setminus \{w\}$ and Q are connected and thus Z' is also a connected dominating set of size at most ℓ for $G \oplus G_1$. This concludes the proof. ◻

Next we prove an auxiliary lemma that upper bounds the number of cut vertices in terms of dominating set of the graph.

Cuts and Blocks. A maximal connected subgraph without a cut vertex is called a **block**. Every block of a graph G is either a maximal 2-connected subgraph, or a bridge or an isolated vertex. By maximality, different blocks of G overlap in at most one vertex, which is then a cut vertex of G. Therefore, every edge of G lies in a unique block and G is the union of its blocks.

Definition 10. *Let A denote the set of cut vertices of G and B the set of its blocks. The bipartite graph on $A \cup B$ where $a \in A$ and $b \in B$ are adjacent when $a \in b$ is called the block graph of G.*

Proposition 2 ([26])**.** *The block graph of a connected graph is a tree.*

Lemma 21. *Let G be a graph and S be a dominating set of G, then the number of cut vertices in G is upper bounded by $|S|$. That is, $|\mathsf{CutVert}| \leq |S|$.*

Proof. Let $A = \mathsf{CutVert}$ denote the set of cut vertices of G and B the set of its blocks. Consider the block graph $\mathcal{B}$ on $A \cup B$. By Proposition 2 we know that $\mathcal{B}$ is a tree. Now we root this tree at some vertex in B. Observe that there is unique association of cut vertices to its parent – which is a block a vertex. We also know that either a cut vertex is in S or a vertex in its parent block is. However, the blocks are pairwise disjoint except for the vertices in A. Thus, this implies that there is an injective map from A to S and hence $|\mathsf{CutVert}| \leq |S|$. $\square$

Now we are ready to prove the treewidth bounding lemma of this section. 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 $8h$-dominating set in the proof. We start with the following auxiliary lemma that will be useful for the proof.

Lemma 22. *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. Also, let $\mathsf{CutVert}$ denote the set of cut vertices of G. We will obtain a $(4h+1)$-dominating set of size $\mathcal{O}(|S|)$ in L_t^*. Towards this end, consider the following set,

$$Q = \bigcup_{A' \subseteq A, A' \text{is feasible}} D(A') \cup R \cup (S \setminus A) \cup \mathsf{CutVert}.$$

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'. By Lemma 21 we have that $|\mathsf{CutVert}| \leq |S|$. Thus $|Q| \leq 2^h(6|S|+6) + 4h \cdot 2^h + 2|S| = \mathcal{O}(|S|)$. 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 $w \in S$ or $w \in \mathsf{CutVert}$ then Q dominates S. So suppose $w \notin R \cup S \cup \mathsf{CutVert}$. 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 $N^{4h}_{G \setminus A}[w]$ which is not in $W(A')$. If $w' \in S_{4h}$, S_{4h} denote the set of vertices in $N^{4h}_{G \setminus A}[S] = N^{4h}_{G \setminus A}[S \setminus A]$, then w is $8h$-dominated by a vertex $w^* \in (S \setminus A) \subseteq Q$ 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 $8h$-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 $8h$-dominating set of a graph, L_t^* has a $8h$-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 $8h$-dominating set of size $\mathcal{O}(|S|)$. By the bidimensionality of $8h$-dominating set, we have that $\mathbf{tw}(L_t^*) = \mathcal{O}(\sqrt{|S|})$ [22, 36]. 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 $\mathbf{tw}(L_t') \leq \mathcal{O}(\sqrt{|S|}) + h = \mathcal{O}(\sqrt{|S|})$. $\square$

Let us also remark that Irrelevant Vertex Rule is based on the performance of a polynomial time approximation algorithm. Thus by Lemmata 2, 20 and 22, and the fact that the treewidth of a graph is at most the maximum treewidth of its torsos, see e.g.[22], we obtain the following lemma.

Lemma 23. *There is a polynomial time algorithm that for a given graph G and a dominating set S of G, outputs graph G' such that $G' \equiv_{\text{CDS}} G$ and for every torso $\tau(t)$ of the tree-decomposition (M, Ψ) of G, we have that $\tau(t) \setminus Z_t$ has a $8h$-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 23 proving Lemma 18 becomes simple.

Proof of Lemma 18. We apply Lemma 23 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 23, $G \setminus A$ has a $8h$-dominating set of size $\delta_3|S|$. Since all vertices of $G \setminus A$ have degree at most h' it follows that $|V(G)| \leq h' + h'^{\mathcal{O}(h')}\delta_3|S| \leq \delta|S|$. $\square$

Proof for Lemma 19 is identical to the proof of Lemma 13, except that we need to use Lemma 23 in place of Lemma 16. Thus we omit it.

Recently, Bodlaender et al. [10] obtained an algorithm solving CDS on graphs of treewidth t in time $c^t n^{\mathcal{O}(1)}$. Theorem 5 combined with this implies that CDS on H-minor-free graphs is solvable in time $2^{\mathcal{O}(\sqrt{k})} + n^{\mathcal{O}(1)}$. To our knowledge, this is the first subexponential parameterized algorithm for CDS on H-minor-free graphs.

Theorem 6. *Given an n-vertex graph G excluding a fixed graph H as a minor, one can check whether G has a connected dominating set of size at most k in time $2^{\mathcal{O}(\sqrt{k})} + n^{\mathcal{O}(1)}$.*

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 kernelization results for DS and CDSbe extended to more general sparse graph classes?

2. Can our techniques be applied to more general families of parameterized problems?

Very recently, the first question was answered both positively and negatively by Drange et al. [30]. In particular, DS admits a vertex-linear kernel on graphs of bounded expansion and an almost vertex-linear kernel on nowhere-dense graphs. On the other hand CDS admits no polynomial kernel on graphs of bounded expansion unless $\mathsf{coNP} \subseteq \mathsf{NP/poly}$.

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 out of reach. Even for the 2-Domination problem on H-minor-free graphs, we are not aware of any polynomial kernelization.

Acknowledgement Thanks to Marek Cygan for sending us a copy of [17].

References

[1] I. Adler, S. G. Kolliopoulos, P. K. Krause, D. Lokshtanov, S. Saurabh, and D. M. Thilikos, *Tight bounds for linkages in planar graphs*, in Proceedings of the 38th International Colloquium of Automata, Languages and Programming (ICALP), vol. 6755 of Lecture Notes in Comput. Sci., Springer, 2011, pp. 110–121.

[2] 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.

[3] J. Alber, M. R. Fellows, and R. Niedermeier, *Polynomial-time data reduction for dominating sets*, J. ACM, 51 (2004), pp. 363–384.

[4] N. Alon and S. Gutner, *Kernels for the dominating set problem on graphs with an excluded minor*, Tech. Rep. TR08-066, ECCC, 2008.

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

[6] N. Alon, P. Seymour, and R. Thomas, *A separator theorem for nonplanar graphs*, J. Amer. Math. Soc., 3 (1990), pp. 801–808.

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

[8] 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.

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

[10] H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof, *Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth*, in Proceedings of the 40th International Colloquium of Automata, Languages and Programming (ICALP), vol. 7965 of Lecture Notes in Comput. Sci., Springer, 2013, pp. 196–207.

[11] 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.

[12] 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.

[13] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos, *(Meta) Kernelization*, CoRR, abs/0904.0727 (2009).

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

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

[16] 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.

[17] M. Cygan, F. Grandoni, and D. Hermelin, *Tight kernel bounds for problems on graphs with small degeneracy - (extended abstract)*, in Proceedings of the 21st Annual European Symposium on Algorithms (ESA), vol. 8125 of Lecture Notes in Comput. Sci., Springer, 2013, pp. 361–372.

[18] 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 Lecture Notes in Comput. Sci., Springer, 2010, pp. 147–158.

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

[20] 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.

[21] 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.

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

[23] 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.

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

[25] 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 Lecture Notes in Comput. Sci., Springer, 2009, pp. 316–327.

[26] R. Diestel, *Graph Theory, 4th Edition*, vol. 173 of Graduate texts in mathematics, Springer, 2012.

[27] 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.

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

[29] F. F. Dragan, F. V. Fomin, and P. A. Golovach, *Spanners in sparse graphs*, J. Comput. Syst. Sci., 77 (2011), pp. 1108–1119.

[30] P. G. Drange, M. Dregi, F. V. Fomin, S. Kreutzer, D. Loksthanov, M. Pilipczuk, M. Pilipczuk, F. Reidl, S. Saurabh, F. S. Villaamil, and S. Sikdar, *Kernelization and sparseness: the case of Dominating Set.* manuscript, 2014.

[31] A. Drucker, *New limits to classical and quantum instance compression*, in Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2012, pp. 609–618.

[32] 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.

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

[34] 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 Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 1989, pp. 520–525.

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

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

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] ——, *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.

[42] 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.

[43] P. A. Golovach and Y. Villanger, *Parameterized complexity for domination problems on degenerate graphs*, in Proceedings of the 34th International wokshop on Graph-Theoretic Concepts in Computer Scienc (WG 2008), vol. 5344 of Lecture Notes in Comput. Sci., Springer, Berlin, 2008, pp. 195–205.

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

[45] M. Grohe and D. Marx, *Structure theorem and isomorphism test for graphs with excluded topological subgraphs*, in Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC), ACM, 2012, pp. 173–192.

[46] 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), Lecture Notes in Comput. Sci., Springer, 2009, pp. 246–257.

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

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

[49] M. Jones, D. Lokshtanov, M. S. Ramanujan, S. Saurabh, and O. Suchý, *Parameterized complexity of directed steiner tree on sparse graphs*, in Proceedings of the 21st Annual European Symposium on Algorithms (ESA), vol. 8125 of Lecture Notes in Comput. Sci., Springer, 2013, pp. 671–682.

[50] 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 Lecture Notes in Comput. Sci., Springer, Berlin, 2008, pp. 47–61.

[51] 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.

[52] 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*, in Proceedings of the 40th International Colloquium of Automata, Languages and Programming (ICALP), vol. 7965 of Lecture Notes in Comput. Sci., Springer, 2013, pp. 613–624.

[53] 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.

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

[55] 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), vol. 5757 of Lecture Notes in Comput. Sci., Springer, 2009, pp. 694–705.

[56] N. Robertson and P. D. Seymour, *Graph minors. XIII. The disjoint paths problem*, J. Comb. Theory Series B, 63 (1995), pp. 65–110.

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

[58] J. M. M. van Rooij, H. L. Bodlaender, and P. Rossmanith, *Dynamic programming on tree decompositions using generalised fast subset convolution*, in Proceedings of the 17th Annual European Symposium on Algorithms (ESA), vol. 5757 of Lecture Notes in Comput. Sci., Springer, 2009, pp. 566–577.