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

Fedor V. Fomin[*1], **Daniel Lokshtanov**[1], **Saket Saurabh**[2], **and Dimitrios M. Thilikos**[†3]

1 **Department of Informatics, University of Bergen, Norway**
{fomin|daniello}@ii.uib.no
2 **The Institute of Mathematical Sciences, Chennai 600113, India.**
saket@imsc.res.in
3 **National and Kapodistrian University of Athens, Greece.**
sedthilk@thilikos.info

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.

1998 ACM Subject Classification G.2.2, F.2.2

Keywords and phrases Parameterized complexity, kernelization, algorithmic graph minors, dominating set, connected dominating set

Digital Object Identifier 10.4230/LIPIcs.STACS.2013.92

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 the 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*.

The Dominating Set (DS) problem together with its numerous variants is one of the most classic and well-studied problems in algorithms and combinatorics [27]. In the Dominating Set problem, we are given a graph G and a non-negative integer k, and the question is whether G contains a set of k vertices whose closed neighborhood contains all the vertices of G. In the connected variant, Connected Dominating Set (CDS), we additionally demand the subgraph induced by the dominating set to be connected. A considerable part of the algorithmic study on these NP-complete problems has been focused on the design of parameterized and kernelization algorithms. In general, DS is W[2]-complete and therefore it cannot be solved by a parameterized algorithm, unless an unexpected collapse occurs in the Parameterized Complexity Hierarchy (see [18]) and thus also does not admit a kernel.

* Supported by European Research Council (ERC) grant "Rigorous Theory of Preprocessing", reference 267959.
† Co-financed by the European Union (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."

30th Symposium on Theoretical Aspects of Computer Science (STACS'13).
Editors: Natacha Portier and Thomas Wilke; pp. 92–103
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

However, there are interesting graph classes where FPT-algorithms exist for the DS problem. The project of widening the horizon where such algorithms exist spanned a multitude of ideas that made DS the testbed for some of the most cutting-edge techniques of parameterized algorithm design. For example, the initial study of parameterized subexponential algorithms for DS on planar graphs [13, 24] resulted in the creation of bidimensionality theory characterizing a broad range of graph problems that admit efficient approximate schemes, fixed-parameter algorithms or kernels on a broad range of graphs [14, 15, 20, 22, 21].

One of the first results on linear kernels is the celebrated work of Alber, Fellows, and Niedermeier on DS on planar graphs [1]. This work augmented significantly the interest in proving polynomial (or preferably linear) kernels for other parameterized problems. The result of Alber et al. [1], see also [8], has been extended to a much more general graph classes like graphs of bounded genus [6] and apex-minor free graphs [22]. An important step in this direction was done by Alon and Gutner [2, 26] who obtained a kernel of size $O(k^h)$ for DS on H-minor-free and H-topological-minor free graphs, where the constant h depends on the excluded graph H. Later, Philip, Raman, and Sikdar [31] 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. In particular, for d-degenerate graphs, a subclass of $K_{i,j}$-free graphs, the algorithm of Philip, Raman, and Sikdar [31] produces a kernel of size $\mathcal{O}(k^{d^2})$. Similarly, the sizes of kernels in [26, 31] are bounded by polynomials in k with degrees depending on the size of the excluded minor H. Alon and Gutner [2] 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 [6], DS has a kernel of size $f(g) \cdot k$ on graphs of genus g. An alternative meta-algorithmic framework, based on bidimensionality theory [14], was introduced in [22], implying the existence of a kernel of size $f(H) \cdot k$ for DS on graphs excluding an apex graph H as a minor. Recently, the result on linear kernels on apex-minor-free graphs was extended to graphs excluding an arbitrary graph H as a minor [23]. 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 [2] for a very special case $H = K_{3,h}$. See Fig. 1 for the relationship between these classes.

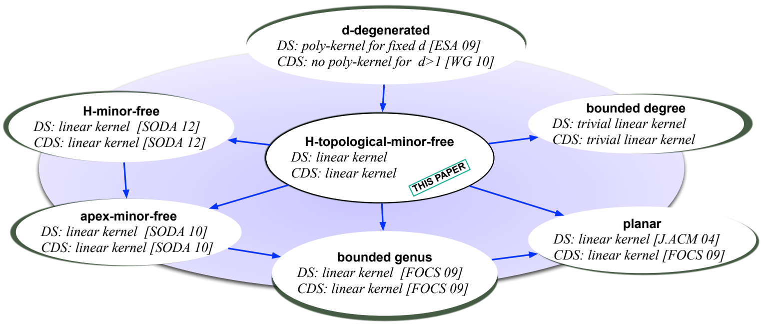

Figure 1 Kernels for DS and CDS on classes of sparse graphs. Arrows represent inclusions of classes (where the class at the head is contained in the class at the tail). In the diagram, [J.ACM 04] is referred to the paper of Albers et al. [1], [FOCS 09] to the paper of Bodlaender et al. [6], [SODA 10] and [SODA 12] to the papers of Fomin et al. [22] and [23], [ESA 09] to the paper of Philip et al. [31], and [WG 10] to Cygan et al. [10].

It is tempting to suggest that similar improvements on kernel sizes are possible for more general graph classes like d-degenerated graphs. For example, for graphs of bounded vertex degree, a subclass of d-degenerate graphs, DS has a trivial linear kernel. Unfortunately, for d-degenerate graphs the existence of a linear kernel and even polynomial kernel with the exponent of the polynomial independent of d is very unlikely. By the very recent work of Cygan et al. [9], the kernelization algorithm of Philip, Raman, and Sikdar [31] is essentially tight—existence of a kernel of size $\mathcal{O}(k^{(d-3)(d-1)-\varepsilon})$, would imply that coNP is in NP/poly. In spite of these negative news, we show how to lift the linearity of kernelization for DS from bounded-degree graphs and H-minor free graphs to the class of graphs excluding H as a topological subgraph. Moreover, a modification of the ideas for DS kernelization can be used to obtain a linear kernel for CDS, which is usually a much more difficult problem to handle due to the connectivity constraint. For example, CDS does not have a polynomial kernel on 2-degenerated graphs unless coNP is in NP/poly [10].

The class of graphs excluding H as a topological subgraph is a wide class of graphs containing H-minor-free graphs and graphs of constant vertex degrees. The existence of a linear kernel for DS on this class of graphs significantly extends and improves previous works [23, 26]. The basic idea behind kernelization algorithms on apex-minor-free and minor-free graphs is the bidimensionality of DS. Roughly speaking, the treewidth of these graphs with dominating set k is either $o(k)$ (as in planar, bounded genus or apex-minor-free graphs [14]) or becomes $o(k)$ after applying the irrelevant vertex technique [23]. This idea can hardly work on graphs of bounded degree, and hence on graphs excluding H as a topological subgraph. The reason is that the bound $o(k)$ on the treewidth of such graphs would imply that DS is solvable in subexponential time on graphs of bounded degree, which in turn can be shown to contradict the Exponential Time Hypothesis [28]. This is why the kernelization techniques developed for H-minor-free graphs does not seem to be applicable directly in our case.

2 Preliminaries

In this section we give various definitions which we make use of in the paper. We refer to Diestel's book [16] for standard definitions from Graph Theory. Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. A graph G' is a *subgraph* of G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. For subset $V' \subseteq V(G)$, the subgraph $G' = G[V']$ of G is called the *subgraph induced by* V' if $E(G') = \{uv \in E(G) \mid u, v \in V'\}$. By $N_G(u)$ we denote the (open) neighborhood of u in graph G. That is, the set of all vertices adjacent to u and by $N[u] = N(u) \cup \{u\}$. Similarly, for a subset $D \subseteq V$, we define $N_G[D] = \cup_{v \in D} N_G[v]$ and $N_G(D) = N_G[D] \setminus D$. We omit the subscripts when they are clear from the context. Throughout the paper, given a graph G and vertex subsets Z and S, whenever we say that a subset Z *dominates all but (everything but)* S then we mean that $V(G) \setminus S \subseteq N[Z]$. Observe that a vertex of S can also be dominated by the set Z.

We denote by K_h the complete graph on h vertices. Also for a given graph G and a vertex subset S, by $K[S]$ we mean a clique on the vertex set S. For an 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 a $u \in P$ such that the distance between u and v is at most r. For $r = 1$, we simply say that Q is dominated by P. We denote by $N_G^r(P)$ the set of vertices r-dominated by P.

Given an edge $e = xy$ of a graph G, the graph G/e is obtained from G by contracting the edge e, that is, the endpoints x and y are replaced by a new vertex v_{xy} which is adjacent to the old neighbors of x and y (except from x and y). A graph H obtained by a sequence of edge-contractions is said to be a *contraction* of G. We denote it by $H \leq_c G$. A graph H is a *minor* of a graph G if H is the contraction of some subgraph of G and we denote it by $H \leq_m G$. We say that a graph G is H*-minor-free* when it does not contain H as a minor.

We also say that a graph class $\mathcal{G}_H$ is *H-minor-free* (or, excludes H as a minor) when all its members are H-minor-free. An *apex graph* is a graph obtained from a planar graph G by adding a vertex and making it adjacent to some of the vertices of G. A graph class $\mathcal{G}_H$ is *apex-minor-free* if $\mathcal{G}_H$ excludes a fixed apex graph H as a minor. A *subdivision* of a graph H is obtained by replacing each edge of H by a path of at least one edge. 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 minor.

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

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

The following notations are the same as that in [25]. Given a tree decomposition of graph $G = (V, E)$, we define mappings $\sigma, \gamma : V(M) \to 2^V$ and $\kappa : E(M) \to 2^V$. For all $t \in V(M)$, $\sigma(t) = \emptyset$ if t is the root of M else $\sigma(t) = \beta(t) \cap \beta(s)$ if s is the parent of t in M. We also set $\gamma(t) = \bigcup_{u \text{ is a descendant of } t} \beta(u)$. For all $e = uv \in E(M)$, $\kappa(e) = \beta(u) \cap \beta(v)$. For a subgraph M' of M by $\beta(M')$ we denote $\cup_{t\in V(M')}\beta(t)$.

Let (M, β) be a tree decomposition of a graph G. The *width* of (M, β) is $min\{|\beta(t)| - 1 \mid t \in V(M)\}$, and the *adhesion* of the tree decomposition is $max\{|\sigma(t)| \mid t \in V(M)\}$. We use $\mathbf{tw}(G)$ to denote the treewidth of the input graph, that is the minimum width of a tree-decomposition of G. For every node $t \in V(M)$, the *torso* at t is the graph $\tau(t) := G[\beta(t)] \cup E(K[\sigma(t)]) \cup \bigcup_{u \text{ child of } t} E(K[\sigma(u)])$.

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

Known Decomposition Theorem. The decomposition theorem that we use extensively for our proofs is given in the next theorem.

▶ **Theorem 1** ([25, 32]). *For every graph H, there exists a constant h, depending only on the size of H, such that for every graph G with $H \not\preceq_T G$, there is a tree decomposition (M, β) of adhesion at most h such that for all $t \in V(M)$, one of the following conditions is satisfied:*

1. *$\tau(t)$ excludes a clique of size h as a minor.*
2. *$\tau(t)$ has at most h vertices of degree at least h (we call these vertices* apices *of $\tau(t)$).*

Moreover, if G is H-minor free graph G then nodes of second type do not exist. Furthermore, there is an algorithm that, given graphs G, H of sizes n and h respectively, computes such a tree decomposition in time $h \cdot n^{O(1)}$ and computes the corresponding apex set Z_t of size at most h for every bag $\tau(t)$.

Actually, we can assume that in (M, β), for any $x, y \in V(M)$, $\beta(x) \not\subseteq \beta(y)$. That is, no bag is contained in other. See [18, Lemma 11.9] for the proof.

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

In this section we give a constant factor approximation for DS on $\mathcal{C}_H$. It is well known that graphs in $\mathcal{C}_H$ have bounded degeneracy. In a recent manuscript a subset of the authors together with others show that DS has a $O(d^2)$ factor approximation algorithm on d-degenerate graphs [29]. To make this paper self contained we provide an approximation algorithm for DS on $\mathcal{C}_H$ here. The main idea of the approximation algorithm is to first compute the tree-decomposition (M, β) given by Theorem 1 for G and then suitably select a bag of this decomposition that still contains a vertex that is not dominated. Then we locally find an approximate dominating set for this bag by using either an approximation algorithm for DS on H-minor free graphs or on graphs of almost bounded degree. We apply this step iteratively and finally show that the dominating set returned by the algorithm is indeed a constant factor approximation. This results in the following lemma.

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

4 Generalized Protrusions

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

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

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

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

Let $\mathcal{G}$ be a class of (not boundaried) graphs. By slightly abusing notation we say that a boundaried graph *belongs in a graph class* $\mathcal{G}$ if the underlying graph belongs in $\mathcal{G}$. By $\partial_G(X)$, we denote the *boundary* of X in G, that is the vertices of G that are not in X and are neighbours of vertices in X.

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

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

Note that for every t, the relation $\equiv_{\Pi,t}$ on t-boundaried graphs is an equivalence relation. Next we define a notion of "transposition-minimality" for the members of each equivalence class of $\equiv_{\Pi,t}$.

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

▶ **Lemma 3** ([6]). *Let* Π *be a parameterized graph problem whose instances are pairs of the form* (G,k) *and let* $t \in \mathbb{Z}^+$*. Then each equivalence class of* $\equiv_{\Pi,t}$ *has a progressive representative.*

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

▶ Definition 7. A parameterized graph problem Π whose instances are pairs of the form (G,k) has *Finite Integer Index* (or simply has *FII*), if and only if for every $t \in \mathbb{Z}^+$, the equivalence relation $\equiv_{\Pi,t}$ is of finite index, that is, has a finite number of equivalence classes. For each $t \in \mathbb{Z}^+$, we define $\mathcal{S}_t$ to be a set containing exactly one progressive representative of each equivalence class of $\equiv_{\Pi,t}$. 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 $\textsc{Thr}(G)$. 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 $\equiv_{\Pi,t}$. For a t-boundaried graph G by $\kappa(G)$ we denote $\max_{G' \in \mathcal{S}_t} \textsc{Thr}(G \oplus G')$.

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

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

5 Slice-Decomposition

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

▶ Definition 8. [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 S is a dominating set of $G[X]$.

The notion of r-DS-protrusion X differs from normal protrusion in the following way. In the normal protrusion we demand that $\mathbf{tw}(X)$ is at most r while in the r-DS-protrusion we demand that the dominating set of the graph induced on X is small. We can similarly define the notion of r-Π-protrusion for various other graph problems Π. The next question is what do we achieve if we get a large r-DS-protrusion (or r-CDS-protrusion). The next lemma shows that in that case we can replace it with an equivalent small graph. More precisely we have the following.

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

Let (M, β) 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 that have exactly one endpoint in $V(M_i)$. Furthermore we define $R_i^+ = \beta(M_i)$ and $\tau(M') := G[R_i^+] \cup \bigcup_{e \in \mathcal{E}(M_i)} K[\kappa(e)]$. Our main objective in this section is to obtain the following (α, β)-*slice decomposition* for $\alpha = \beta = \mathcal{O}(k)$.

▶ Definition 9. [(α, β)**-slice decomposition**] Let G be a graph with $H \not\preceq_T G$ and let (M, β) be the tree decomposition given by Theorem 1. An (α, β)-*slice decomposition* of a graph G is a collection $\mathcal{P}$ of pairwise disjoint connected subtrees $\{M_1, \ldots, M_\alpha\}$ of M such that the following holds.

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

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

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

▶ Definition 10. Let (M, β) be the tree decomposition of a graph G given by the Theorem 1. For a subset $Q \subseteq V(G)$ and a subtree M' of M we define $\mu(M', Q) = |\beta(M') \cap Q|$. 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.

▶ **Lemma 6.** *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 the problem* DS *(*CDS*)) and a polynomial time algorithm such that given a yes instance (G, k) of* DS *(*CDS*), 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;*
- *a h'-protrusion of size more than $\xi_{h'}$ where h' depends only on h.*

Sketch of the proof. To obtain the slice-decomposition we introduce our marking scheme as follows.

1. Apply Lemma 2 on the input graph G and compute a $\eta(H)$-factor approximation (connected) dominating set D for G.
2. Use Theorem 1 and compute a tree-decomposition (M, β). 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 all the set of edges that have been marked.

Let M^* be the subtree (requires proof) induced on all the heavy edges. We use this tree M^* to obtain the decomposition. If (G, k) is a yes instance then one can show that the number of leaves in the tree M^* is upper bounded by $\mathcal{O}(k)$. This immediately implies that the number of maximal paths consisting only of degree 2 vertices is upper bounded by $\mathcal{O}(k)$. We show that if any of these paths is too long then we can obtain a $2h$-DS-protrusion of large size. This implies that the size of the tree M^* is upper bounded by $\mathcal{O}(k)$. Now we delete all the edges appearing in M^* from M. This breaks the tree M into $\mathcal{O}(k)$ subtrees, $\mathcal{P} = \{M_1, \ldots, M_\alpha\}$. We argue that these subtrees form the partition described in the definition of slice decomposition. To show that $\sum_{i=1}^{\rho}(\sum_{e \in \mathcal{F}(M_i)} |\kappa(e)|) \leq \mathcal{O}(k)$, we use the fact that any edge of M^* sees at most two trees among $\mathcal{P}$. ◀

6 Final Kernel

In this section we use slice-decomposition obtained in the last section and the reduction rules used in [23] to obtain linear kernels for DS and CDS. We first outline our algorithm for DS and then explain how we can obtain a linear kernel for CDS.

Kernelization Algorithm for DS**.** 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 6 and

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

In the second case we apply Lemma 5. Given X we apply Lemma 5 and obtain a boundaried graph X' such that $|X'| \leq \xi_{2h}$ and $X \equiv_{\text{DS},2h} X'$ ($X \equiv_{\text{CDS},2h} X'$). We also compute the translation constant c between X and X'. Now we replace the graph X with X' and obtain a new equivalent instance $(G', k + c)$. (Recall that c is a non-positive integer). In the third case we apply the protrusion replacement lemma of [6, 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 6 returns a slice-decomposition. For simplicity we denote by (G, k) itself the graph on which Lemma 6 returns the slice-decomposition. Since the number of times this process can be repeated is upper bounded by $n = |V(G)|$, we can obtain $(\delta_1 k, \delta_2 k)$-slice decomposition for (G, k) in polynomial-time.

Let $\mathcal{P}$ be the pairwise disjoint connected subtrees $\{M_1, \ldots, M_\alpha\}$ of M coming from the slice-decomposition of G. Recall that $R_i^+ = \beta(M_i)$. 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 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 lemmata.

▶ **Lemma 7.** *There exists a constant δ and a polynomial time algorithm that, given a graph G with boundary S where S is a dominating set for G and G has at most h' vertices of degree at least h', outputs a graph G' with boundary S such that $G' \equiv_{\text{DS},|S|} G$ and $|V(G')| \leq \delta|S|$. Furthermore we can also compute the translation constant c of G and G' in polynomial-time.*

▶ **Lemma 8.** *There exists a constant δ and a polynomial time algorithm that, given an H-minor free graph G with boundary S where S is a dominating set for G, outputs a graph G' with boundary S such that $G' \equiv_{\text{DS},|S|} G$ 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 7 and 8, we obtain the linear sized kernel for DS as follows. Given the graph G we obtain the slice-decomposition and check if any of G_i has size more than δb_i. If yes then we either apply Lemma 7 or Lemma 8 based on the type of G_i and obtain a graph G_i' such that $G_i' \equiv_{\text{DS},b_i} G_i$ and $|V(G_i')| \leq \delta b_i$. We think $G = G_i \oplus G^\star$, where $G^\star = G \setminus (R_i^+ \setminus B_i)$ as a b_i-boundaried graph with boundary B_i. Then we obtain a smaller equivalent graph $G' = G^\star \oplus G_i'$ and $k' = k + c$. After this we can repeat the whole process once again. This implies that when we can not apply Lemmata 8 or 7 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: $\sum_{i=1}^{\alpha} |R_i^+| \leq \delta \sum_{i=1}^{\alpha} b_i = \delta(\sum_{i=1}^{\alpha}(|Q_i| + |(D \cap R_i^+) \setminus Q_i|)) = \delta(\sum_{i=1}^{\alpha} |Q_i| + \sum_{i=1}^{\alpha} |(D \cap R_i^+) \setminus Q_i|) \leq \delta\delta_2 k + \delta\eta(H)k = \mathcal{O}(k)$. This bring us to the following theorem.

▶ **Theorem 9.** DS *admits a linear kernel on graphs excluding a fixed graph H as a topological minor.*

It only remains to prove Lemmata 7 and 8 to complete the proof of Theorem 9.

Irrelevant Vertex Rule and proof for Lemma 7. For the proofs of Lemmata 7 and 8 we need to use an irrelevant vertex rule developed in [23]. Furthermore, the proof of Lemma 8 is essentially a reformulation of the results presented in [23].

If the graph G is $K_{h'}$-minor free then the irrelevant vertex rule will be used in a recursive fashion. In each recursive step it is used in order to reduce the treewidth of torsos and hence also the entire graph. Then the graph is split in two pieces and the procedure is applied recursively to the two pieces. In the bottom of the recursion when the graph becomes smaller but still big enough then we apply Lemma 5 on it and obtain an equivalent instance.

Let G be a graph given with its tree-decomposition (M, β) 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 [20] 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 10.** *Let S be a dominating set in G, and G' be the graph obtained by applying the Irrelevant Vertex Rule on G, where w was the deleted vertex. Then $G' \equiv_{\mathrm{DS},|S|} G$.*

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

Now, let $Z' \subseteq V(G' \oplus G_1)$ be a dominating set of size at most ℓ for $G' \oplus G_1$. Let $Z_1' = V(G') \cap Z'$. As in the forward direction we can assume that $|Z_1'| \leq |S|$. We show that Z' also dominates w in $G \oplus G_1$. Specifically $Z_1' \cup \{w\}$ is a dominating set of all but S in G of size at most $|S| + 1$ so $Z_1' \cap A$ is feasible. Since $\{w\}$ is irrelevant with respect to $Z_1' \cap A$, we have $w \in N_G(Z_1' \cap A)$ and thus Z' is a dominating set for $G' \oplus G_1$ of size at most ℓ. This concludes the proof. ◀

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

▶ **Lemma 11.** *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, β) 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 of Lemma 7. We apply Lemma 11 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 11, $G \setminus A$ 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 $|V(G)| \leq h' + \delta_3 h|S|\delta_3 h^2|S| \leq \delta|S|$. ◀

Kernelization algorithm for CDS**.** To obtain kernelization algorithm for CDS the only thing that remains to show are results analogous to Lemmata 8 and 7 for DS. However to obtain this we need to apply reduction rules developed in [23] for CDS. Finally we need to adapt the proofs of Lemmata 11, 12, 13 and 14 given in the full-version available at [23] with the new perspective. Two of these lemmata essentially shows the correctness of reduction rules for CDS and that every torso has 2-dominating set of size at most $\mathcal{O}(|S|)$. Here S is a connected dominating set of the input graph G. The only result that is not proved in [23] is the result analogous to Lemma 7 for DS. However, the size of a dominating set is at most the size of a connected dominating set. After this the proof for the case that given a graph G with at most h' vertices of degree at least h' we can return a canonically equivalent graph G' is verbatim to the proof of Lemma 7. We omit these adaptation details from this extended abstract.

▶ **Theorem 12.** CDS *admits a linear kernel on graphs excluding a fixed graph H as a topological minor.*

7 Conclusions

In this paper we give linear kernels for two widely studied parameterized problems, namely DS and CDS, for every graph class that excludes some graph as a topological minor. The emerging questions are the following two: (**1**) Can our techniques be extended to more general sparse graph classes? (**2**) Can our techniques be applied to more general families of parameterized problems? We believe that any step towards resolving the first question should be based on significant graph-theoretical advances. Our results make use of the decomposition theorem of Grohe and Marx in [25] that, in turn, can be seen as an extension of seminal results of the Graph Minor Series by Robertson and Seymour [32]. So far no similar structural theorem is known for more general sparse graph classes. We also believe that a broadening of the kernelization horizon for these two problems without the use of some tree-based structural characterization of sparsity requires completely different ideas.

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

References

1 J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating sets. *J. ACM*, 51:363–384, 2004.

2 N. Alon and S. Gutner. Kernels for the dominating set problem on graphs with an excluded minor. Technical Report TR08-066, ECCC, 2008.

3 S. Arnborg, B. Courcelle, A. Proskurowski, and D. Seese. An algebraic theory of graph reduction. *Journal of the ACM*, 40:1134–1164, 1993.

4 H. L. Bodlaender and B. de Fluiter. Reduction algorithms for constructing solutions in graphs with small treewidth. pages 199–208, 1996.

5 H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

6 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)*, pages 629–638. IEEE, 2009.

7 H. L. Bodlaender and B. van Antwerpen-de Fluiter. Reduction algorithms for graphs of small treewidth. *Information and Computation*, 167:86–119, 2001.

8 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.*, 37:1077–1106, 2007.

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

10 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)*, volume 6410 of *Lect. Notes Comp. Sc.*, pages 147–158. Springer, 2010.

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

12 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)*, pages 251–260, 2010.

13 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(1):33–47, 2005.

14 E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *J. ACM*, 52(6):866–893, 2005.

15 E. D. Demaine and M. Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51(3):332–337, 2007.

16 R. Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005.

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

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

19 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)*, volume 9 of *LIPIcs*, pages 189–200. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

20 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)*, pages 748–759. SIAM, 2010.

21 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)*, pages 1563–1575. SIAM, 2012.

22 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)*, pages 503–510. ACM-SIAM, 2010.

23 F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Linear kernels for (connected) dominating set on H-minor-free graphs. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, http://www.ii.uib.no/ daniello/.

24 F. V. Fomin and D. M. Thilikos. Dominating sets in planar graphs: Branch-width and exponential speed-up. *SIAM J. Comput.*, 36:281–309, 2006.

25 M. Grohe and D. Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. In *STOC*, pages 173–192, 2012.

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

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

28 R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

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

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

31 G. Philip, V. Raman, and S. Sikdar. Solving dominating set in larger classes of graphs: FPT algorithms and polynomial kernels. In *Proceedings of the 17th Annual European Symposium on Algorithms (ESA 2009)*, volume 5757 of *Lect. Notes Comp. Sc.*, pages 694–705. Springer, 2009.

32 N. Robertson and P. D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *J. Combin. Theory Ser. B*, 89(1):43–76, 2003.