Fixed-parameter Tractable Distances to Sparse Graph Classes

Jannis Bulian and Anuj Dawar

University of Cambridge Computer Laboratory, UK
firstname.lastname@cl.cam.ac.uk

Abstract

We show that for various classes $\mathcal{C}$ of sparse graphs, and several measures of distance to such classes (such as edit distance and elimination distance), the problem of determining the distance of a given graph G to $\mathcal{C}$ is fixed-parameter tractable. The results are based on two general techniques. The first of these, building on recent work of Grohe et al. establishes that any class of graphs that is slicewise nowhere dense and slicewise first-order definable is FPT. The second shows that determining the elimination distance of a graph G to a minor-closed class $\mathcal{C}$ is FPT.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases parameterized complexity, fixed-parameter tractable, distance, graph theory, sparse graphs, graph minor, nowhere dense

Digital Object Identifier 10.4230/LIPIcs.IPEC.2015.236

1 Introduction

The study of parameterized algorithmics for graph problems has thrown up a large variety of structural parameters of graphs. Among these are parameters that measure the *distance* of a graph G to a class $\mathcal{C}$ in some way. The simplest such measures are those that count the number of vertices or edges that one must delete (or add) to G to obtain a graph in $\mathcal{C}$. A common motivation for studying such parameters is that if a problem one wishes to solve is tractable on the class $\mathcal{C}$, then the distance to $\mathcal{C}$ provides an interesting parameterization of that problem (called *distance to triviality* by Guo et al. [12]). Other examples of this include the study of modulators to graphs of bounded tree-width in the context of kernelization (see [8, 9]) or the parameterizations of colouring problems (see [15]). On the other hand, determining the distance of an input graph G to a class $\mathcal{C}$ is, in general, a computationally challenging problem in its own right. Such problems have also been extensively studied with a view to establishing their complexity when parameterized by the distance. A canonical example is the problem of determining the size of a minimum vertex cover in a graph G, which is just the vertex-deletion distance of G to the class of edge-less graphs. More generally, Cai [3] studies the parameterized complexity of distance measures defined in terms of addition and deletion of vertices and edges to hereditary classes $\mathcal{C}$. Counting deletions of vertices and edges gives a rather simple notion of distance, and many more involved notions have also been studied. Classic examples include the crossing number of a graph which provides one notion of distance to the class of planar graphs or the treewidth of a graph which can be seen as a measure of distance to the class of trees. Another recently introduced measure is *elimination distance*, defined in [2] where it was shown that graph isomorphism is FPT when parameterized by elimination distance to a class of graphs of bounded degree.

In this paper we consider the fixed-parameter tractability of a variety of different notions of distance to various different classes $\mathcal{C}$ of sparse graphs. We establish two quite general

10th International Symposium on Parameterized and Exact Computation (IPEC 2015).
Editors: Thore Husfeldt and Iyad Kanj; pp. 236–247
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

techniques for establishing that such a distance measure is FPT. The first builds on the recent result of Grohe et al. [11] which shows that the problem of evaluating first-order formulas on any *nowhere dense* class of graphs is FPT with the formula as parameter. We extract from their proof of this result a general statement about the fixed-parameter tractability of definable sparse classes. To be precise, we show that parameterized problems that are both *slicewise nowhere dense* and *slicewise first-order definable* (these terms are defined precisely below) are FPT. As an application of this, it follows that if $\mathcal{C}$ is a nowhere dense class of graphs that is definable by a first-order formula, then the parameterized problem of determining the distance of a graph G to $\mathcal{C}$ is FPT, for various notions of distance that can be themselves so defined. In particular, we get that various forms of edit distance to classes of bounded-degree graphs are FPT (a result established by Golovach [10] by more direct methods). Another interesting application is obtained by considering elimination distance of a graph G to the class $\mathcal{C}$ of empty graphs. This is nothing other than the *tree-depth* of G. While elimination distance to a class $\mathcal{C}$ is in general not first-order definable, it is in the particular case where $\mathcal{C}$ is the class of empty graphs. Thus, we obtain as an application of our method the result that tree-depth is FPT, a result previously known from other algorithmic meta theorems (see [16, Theorem 17.2]). The method of establishing that a parameterized problem is FPT by establishing that it is slicewise nowhere dense and slicewise first-order definable appears to be a powerful method of some generality which will find application beyond these examples.

Our second general method specifically concerns elimination distance to a minor-closed class $\mathcal{C}$. We show that this measure is fixed-parameter tractable for any such $\mathcal{C}$, answering an open question posed in [2]. Note that while a proper minor-closed class is always nowhere dense, it is not generally first-order definable (for instance, neither the class of acyclic graphs nor the class of planar graphs is), and elimination distance to such a class is also not known to be first-order definable. Thus, our results on the tractability of slicewise first-order definable classes do not apply here. Instead, we build on work of Adler et al. [1] to show that from a finite list of the forbidden minors characterising $\mathcal{C}$, we can compute the set of forbidden minors characterising the graphs at elimination distance k to $\mathcal{C}$. Adler et al. show how to do this for apex graphs, from which one immediately obtains the result for graphs that are k deletions away from $\mathcal{C}$. To extend this to elimination distance k, we show how we can construct the forbidden minors for the closure of a minor-closed class under disjoint unions.

In Section 2 we present the definitions necessary for the rest of the paper. Section 3 establishes our result for slicewise first-order definable and slicewise nowhere dense problems and gives some applications of the general method. Section 4 establishes that the problem of determining elimination distance to any minor-closed class is FPT. Some open questions are discussed in Section 5. Due to limitations of space, some material is deferred to the full version of this paper, which may be found at `arxiv:1502.05910`.

2 Preliminaries

First-order logic

We assume some familiarity with first-order logic for Section 3. A *(relational) signature* σ is a finite set of relation symbols, each with an associated arity. A σ*-structure* A consists of a set $V(A)$ and for each k-ary relation symbol $R \in \sigma$ a relation $R(A) \subseteq V(A)^k$. Our structures are mostly (coloured) graphs, so $\sigma = \{E\}$ or $\sigma = \{E, C_1, C_2, \dots, C_r\}$ where E is binary and the C_i are unary relation symbols. A graph G is then a σ-structure with vertex set $V(G)$, edge relation $E(G)$, and colours $C_i(G)$.

A first-order formula φ is recursively defined by the following rules:

$$\varphi := R(x_1, \ldots, x_r) \mid x = y \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\varphi.$$

We also use the following abbreviations:

$$\varphi \wedge \psi := \neg(\neg\varphi \vee \neg\psi), \qquad \forall x.\varphi := \neg\exists.\neg\varphi.$$

The *quantifier rank* of a formula φ is the nesting depth of quantifiers in φ. For a more detailed presentation we refer to Hodges [13].

Parameterized Complexity

Parameterized complexity theory is a two-dimensional approach to the study of the complexity of computational problems. We find it convenient to define problems as classes of structures rather than strings. A *problem* $Q \subseteq str(\sigma)$ is an (isomorphism-closed) class of σ-structures given some signature σ. A *parameterization* is a computable function $\kappa : str(\sigma) \to \mathbb{N}$. We say that Q is *fixed-parameter tractable* with respect to κ if we can decide whether an input $A \in str(\sigma)$ is in Q in time $O(f(\kappa(A)) \cdot |A|^c)$, where c is a constant and f is some computable function. For a thorough discussion of the subject we refer to the books by Downey and Fellows [5], Flum and Grohe [7] and Niedermeier [17].

A parameterized problem (Q, κ) is *slicewise first-order definable* if there is a computable function $f : \mathbb{N} \to \mathrm{FO}[\sigma]$ such that a σ-structure A with $\kappa(A) \leq i$ is in Q if, and only if, $A \models f(i)$. Slicewise definability of problems in a logic was introduced by Flum and Grohe [6].

Graph theory

A *graph* G is a set of vertices $V(G)$ and a set of edges $E(G) \subseteq V(G) \times V(G)$. We assume that graphs are loop-free and undirected, i.e. that E is irreflexive and symmetric. We mostly follow the notation in Diestel [4]. For a set $S \subseteq V(G)$ of vertices, we write $G \setminus S$ to denote the subgraph of G induced by $V(G) \setminus S$.

Let $r \in \mathbb{N}$. An r-independent set in a graph G is a set of vertices of G such that their pairwise distance is at least r.

A graph H is a *minor* of a graph G, written $H \preceq G$, if there is a map, called the *minor map*, that takes each vertex $v \in V(H)$ to a tree T_v that is a subgraph of G such that for any $u \neq v$ the trees are disjoint, i.e. $T_v \cap T_u = \emptyset$, and such that for every edge $uv \in E(H)$ there are vertices $u' \in T_u, v' \in T_v$ with $u'v' \in E(G)$. A class of graphs $\mathcal{C}$ is *minor-closed* if $H \preceq G \in \mathcal{C}$ implies $H \in \mathcal{C}$.

The *set of minimal excluded minors* $M(\mathcal{C})$ is the set of graphs in the complement of $\mathcal{C}$ such that for each $G \in M(\mathcal{C})$ all proper minors of G are in $\mathcal{C}$. By the Robertson-Seymour Theorem [18] the set $M(\mathcal{C})$ is finite for every minor-closed class $\mathcal{C}$. It is a consequence of this theorem that membership in a minor-closed class can be tested in $O(n^3)$ time. For a set M of graphs, we write $\mathrm{Forb}(M)$ for the class of graphs which forbid M as minors, i.e. $\mathrm{Forb}(M) = \{G \mid H \not\preceq G \text{ for all } H \in M\}$.

Let $r \in \mathbb{N}$. A minor H of G is a *depth-r minor of G*, written $H \preceq_r G$, if there is a minor map that takes vertices in H to trees that have radius at most r. A class of graphs $\mathcal{C}$ is *nowhere dense* if for every $r \in \mathbb{N}$ there is a graph H_r such that for no $G \in \mathcal{C}$ we have $H_r \preceq_r G$. A nowhere-dense class of graphs $\mathcal{C}$ is called *effectively nowhere dense* if there is a computable function f from integers to graphs such that for no $G \in \mathcal{C}$ and no r we have $f(r) \preceq_r G$. We are only interested in effectively nowhere-dense classes so we simply use the term *nowhere dense* to mean effectively nowhere dense.

We say that a parameterized graph problem (Q, κ) is *slicewise nowhere dense* if there is a computable function h from pairs of integers to graphs such that for all $i \in \mathbb{N}$, we have for no $G \in \{H \in Q \mid \kappa(H) \leq i\}$ and r that $h(i,r) \preceq_r G$. We will call h the *parameter function* of Q.

For a class of graphs $\mathcal{C}$ we denote the closure of $\mathcal{C}$ under taking disjoint unions by $\overline{\mathcal{C}}$. We say that a graph G is an *apex graph* over a class $\mathcal{C}$ of graphs if there is a vertex $v \in V(G)$ such that the graph $G \setminus \{v\} \in \mathcal{C}$. The class of all apex graphs over $\mathcal{C}$ is denoted $\mathcal{C}^{\text{apex}}$.

A graph G has *deletion distance k to a class $\mathcal{C}$* if there are k vertices $v_1, \ldots, v_k \in V(G)$ such that $G \setminus \{v_1, \ldots, v_k\} \in \mathcal{C}$.

The *elimination distance* of a graph G to a class $\mathcal{C}$ is defined as follows:

$$ed_{\mathcal{C}}(G) := \begin{cases} 0, & \text{if } G \in \mathcal{C}; \\ 1 + \min\{ed_{\mathcal{C}}(G \setminus v) \mid v \in V(G)\}, & \text{if } G \notin \mathcal{C} \text{ and } G \text{ is connected;} \\ \max\{ed_{\mathcal{C}}(H) \mid H \text{ a connected component of } G\}, & \text{otherwise.} \end{cases}$$

3 A general method for editing distances

In this section we establish a general technique for showing that certain definable parameterized problems on graphs are FPT. As an application, we show that certain natural distance measures to sparse graph classes are FPT. To be precise, we show that if a parameterized problem is both slicewise first-order definable and slicewise nowhere dense, then it is FPT. In particular, this implies that if we have a class $\mathcal{C}$ that is first-order definable and nowhere dense and the distance measure we are interested in is also first-order definable (that is to say, for each k there is a formula that defines the graphs of distance k from $\mathcal{C}$), then the problem of determining the distance is FPT. More generally, if we have a parameterized problem (Q, κ) that is slicewise nowhere dense and slicewise first-order definable, and a measure of distance to it is definable in the sense that for any values of k and d, there is a first-order formula defining the graphs of distance d to the class $\{G \mid G \in Q \text{ and } \kappa(G) \leq k\}$, then the problem of deciding whether a graph has distance at most d to this class is FPT parameterized by $d+k$. In particular, this yields the result of Golovach [10] as a consequence.

The method is an adaption of the main algorithm in Grohe *et al.* [11]. Since the proof is essentially a modification of their central construction, rather than give a full account, we state the main results they prove and explain briefly how the proofs can be adapted for our purposes. For a full proof, this section is best read in conjunction with the paper [11]. Section 3.1 gives an overview of the key elements of the construction from [11] and the elements from it which we need to extract for our result. Section 3.2 then gives our main result and Section 3.3 derives some consequences for distance measures.

3.1 Evaluating Formulas on Nowhere Dense Classes

The key result of [11] is:

▶ **Theorem 1** (Grohe *et al.* [11, Theorem 1.1])**.** *For every nowhere dense class $\mathcal{C}$ and every $\epsilon > 0$, every property of graphs definable in first-order logic can be decided in time $O(n^{1+\epsilon})$ on $\mathcal{C}$.*

In the full version of the paper we give a sketch of the algorithm from Theorem 1 with an emphasis on the changes needed for our purposes. Here we state the main results that we extracted and that are needed for the next section.

A key data structure used in the algorithm is a neighbourhood cover. An important result from [11] is that graphs from a nowhere dense class allow for small covers and that such a cover can be efficiently computed.

▶ **Theorem 2** (Grohe *et al.* [11, Theorem 6.2]). *Let $\mathcal{C}$ be a nowhere dense class of graphs. There is a function f such that for all $r \in \mathbb{N}$ and $\epsilon > 0$ and all graphs $G \in \mathcal{C}$ with $n \geq f(r, \epsilon)$ vertices, there exists an r-neighbourhood cover of radius at most $2r$ and maximum degree at most n^ϵ and this cover can be computed in time $f(r, \epsilon) \cdot n^{1+\epsilon}$. Furthermore, if $\mathcal{C}$ is effectively nowhere dense, then f is computable.*

While the algorithm of [11] assumes that the input graph G comes from the class $\mathcal{C}$, we can say something more. For a fixed nowhere dense class $\mathcal{C}$, where we know the parameter function h, we can, given G, r and ϵ, compute a bound on the running time of the algorithm from Theorem 2 . By running the algorithm to this bound, we have the following as a direct consequence.

▶ **Lemma 3.** *Let $\mathcal{C}$ be a nowhere dense class of graphs. There is a function f such that for all $r \in \mathbb{N}$ and $\epsilon > 0$ and all graphs $G \in \mathcal{C}$ with $n \geq f(r, \epsilon)$ vertices, there exists an r-neighbourhood cover of radius at most $2r$ and maximum degree at most n^ϵ. There is an algorithm that given an arbitrary graph G runs in time $f(r, \epsilon) \cdot n^{1+\epsilon}$ and that computes this cover or determines that $G \notin \mathcal{C}$. Furthermore, if $\mathcal{C}$ is effectively nowhere dense, then f is computable.*

At the core of the proof of Theorem 1 is the Rank-Preserving Locality Theorem. We state a simplified version here. More details can be found in the full version of the paper.

▶ **Theorem 4** (Rank-Preserving Locality Theorem, Grohe *et al.* [11, Theorem 7.5]). *For every $q \in \mathbb{N}$ there is an r such that for every FO-formula $\varphi(x)$ of quantifier rank q there is a formula with an extended signature $\hat{\varphi}(x)$ and a graph G' (both depending on q and r) such that for every $v \in V(G)$,*

$$G \models \varphi(v) \iff G' \models \hat{\varphi}(v).$$

Furthermore, $\hat{\varphi}$ is computable from φ, and r is computable from q.

We observe that the structure G' mentioned in Theorem 4 can be efficiently computed:

▶ **Lemma 5.** *Let $\mathcal{C}$ be a nowhere dense class of graphs. There is an algorithm that runs in time $O(q)$ which, given a graph G, returns G' or determines that $G \notin \mathcal{C}$.*

Theorem 4 reduces the problem of evaluating a formula of first-order logic to deciding a series of distance-r-independent set problems. So, the final ingredient is to show that this is tractable. Formally, the problem is defined as follows:

DISTANCE INDEPENDENT SET
Input: A graph G and $k, r \in \mathbb{N}$.
Parameter: $k + r$
Problem: Does G contain an r-independent set of size k?

The problem is shown to be FPT on nowhere dense classes of graphs [11, Theorem 5.1], and the theorem can be restated as follows:

▶ **Lemma 6.** *Let $\mathcal{C}$ be a nowhere dense class of graphs. Then there is an algorithm and a function f such that for every $\epsilon > 0$ the algorithm runs in time $f(\epsilon, r, k)$ and either solves the* DISTANCE INDEPENDENT SET *problem or determines $G \notin \mathcal{C}$. If $\mathcal{C}$ is effectively nowhere dense, then f is computable.*

This is all we need to evaluate $\hat{\varphi}$ on G', which is equivalent to evaluating φ on G by Theorem 4.

3.2 Deciding Definable nowhere dense Problems

The main result of [11] establishes that checking whether $G \models \varphi$ is FPT when parameterized by φ provided that G comes from a known nowhere dense class $\mathcal{C}$. Thus, the formula is arbitrary, but the graphs come from a restricted class. Section 3.1 gives an account of this proof from which we can extract the observation that the algorithm can be modified to work for an arbitrary input graph G with the requirement that the algorithm *may* simply reject the input if G is not in $\mathcal{C}$. This suggests a tractable way of deciding $G \models \varphi$ provided that φ defines a nowhere dense class. Now the graph is arbitrary, but the formula comes from a restricted class. We formalise the result in the following theorem:

▶ **Theorem 7.** *Let (Q, κ) be a problem that is slicewise first-order definable and slicewise nowhere dense. Then (Q, κ) is fixed-parameter tractable.*

Proof. In the following, for ease of exposition, we assume that an instance of the problem consists of a graph G and $\kappa(G) = i$ for some positive integer i.

Step 1: Compute φ and the parameters function: Since (Q, κ) is slicewise first-order definable, we can compute from i a first-order formula φ which defines the class of graphs $C_i = \{H \mid H \in Q \text{ and } \kappa(H) \leq i\}$. Moreover, since (Q, κ) is slicewise nowhere dense, we can compute from i an algorithm that computes the parameter function h for C_i.

Step 2: Obtain $\hat{\varphi}$ from φ: By the Rank-Preserving Locality Theorem (Theorem 4), we can compute from φ the formula $\hat{\varphi}$ and a radius r.

Step 3: Find a small cover $\mathcal{X}$ for G: By Lemma 3, we can either find a cover $\mathcal{X}$ for G, or reject if the algorithm determine that $G \notin C_i$.

Step 4: Simulate Splitter game to compute G': By Lemma 5 we obtain G' or reject if the algorithm determines that $G \notin C_i$.

Step 5: Evaluate $\hat{\varphi}$ on G': Finally to evaluate $\hat{\varphi}$ on G', we need to solve the distance independent set problem. We can do this by Lemma 6. Since evaluating $\hat{\varphi}$ on G' is equivalent to evaluating φ on G this allows us to decide whether $G \in Q$. ◀

3.3 Applications

In this Section we discuss some applications of Theorem 7 that demonstrate its power. We begin by considering simple edit distances.

Edit Distances

A graph G has *deletion distance* k to a class $\mathcal{C}$ if there exists a set S of k vertices in G so that $G \setminus S \in \mathcal{C}$. Suppose (Q, κ) is a parameterized graph problem. We define the problem of deletion distance to Q as follows:

Deletion Distance to Q
Input: A graph G and $k, d \in \mathbb{N}$.
Parameter: $k + d$
Problem: Does G contain a set S of k vertices so that $\kappa(G \setminus S) \leq d$ and $G \setminus S \in Q$?

In many of the examples below, we define formulas of first-order logic by *relativisation*. For convenience, we define the notion here.

▶ **Definition 8.** Let φ and $\psi(x)$ be first-order formulas, where ψ has a distinguished free variable x . The relativisation of φ by ψ, denoted $\varphi^{[x.\psi]}$ is the formula obtained from φ by replacing all subformulas of the form $\exists v\varphi'$ in φ by $\exists v(\psi[v/x] \wedge \varphi')$, and all subformulas of the form $\forall v\varphi'$ in φ by $\forall v(\psi[v/x] \rightarrow \varphi')$. Here $\psi[v/x]$ denotes the result of replacing the free occurrences of x in ψ with v in a suitable way avoiding capture.

The key idea here is that $\varphi^{[x.\psi]}$ is true in G iff φ is true in the subgraph of G induced by the vertices that satisfy $\psi(x)$. Note that the variable x that is free is ψ is bound in $\varphi^{[x.\psi]}$. Other variables that appear free in ψ remain free in $\varphi^{[x.\psi]}$.

▶ **Proposition 9.** *If (Q, κ) is slicewise nowhere dense and slicewise first-order definable then* Deletion Distance to Q *is* FPT.

Proof. It suffices to show that Deletion Distance to Q is also slicewise nowhere dense and slicewise first-order definable. For the latter, note that if φ_i is the first-order formula that defines the class of graphs $\mathcal{C}_i = \{G \mid \kappa(G) \leq i \text{ and } G \in Q\}$, then the class of graphs at deletion distance k to $\mathcal{C}_i$ is given by:

$$\exists w_1, \ldots, w_k \varphi_i^{[x.\theta_k]}$$

where $\theta_k(x)$ is the formula $\bigwedge_{1 \leq i \leq k} x \neq w_i$.

Since $\mathcal{C}_i$ is slicewise nowhere dense, there is a computable function f such that for all i and r the graph $H_i = f(i)$ is not an r-minor of any of the graphs in the class $\mathcal{C}_i$. Observe that if a graph $G \in \mathcal{C}_i$ excludes H_i with $|H_i| = m_i$ vertices as an r-minor, then it also excludes K_{m_i} as an r-minor.

To see that Deletion Distance to Q is also slicewise nowhere dense, note that a graph with deletion distance k to a graph in $G \in \mathcal{C}_i$ cannot contain K_{m_i+k} as an r-minor. We can thus define $g(r, k) = K_{m_i+k}$ as the parameter function of the class of graphs with deletion distance k to $\mathcal{C}_i$. ◀

Instead of deleting vertices, we can also consider editing the graph by adding or deleting edges. It is easily seen that we can modify a first-order formula φ to define the class of graphs G that can be made to satisfy G by k edge additions or deletions. Thus, an analogue of Proposition 9 is obtained for any combination of vertex and edge deletions and additions. Golovach [10] proved that that editing a graph to degree d using at most k edge additions/deletions is FPT parameterized by $k + d$. Since the class of graphs of degree d is first-order definable and nowhere dense for any d, the result also now follows from Theorem 7.

Tree-depth

Tree-depth is a graph parameter that lies between the widely studied parameters vertex cover number and tree width. It has interesting connections to nowhere dense graph classes. It is usually defined as follows:

▶ **Definition 10.** The *tree-depth* of a graph G, written $td(G)$, is

$$td(G) := \begin{cases} 0, & \text{if } V(G) = \emptyset; \\ 1 + \min\{td(G \setminus v) \mid v \in V(G)\}, & \text{if } G \text{ is connected;} \\ \max\{td(H) \mid H \text{ a component of } G\}, & \text{otherwise.} \end{cases}$$

Note that a graph has tree-depth k if, and only if, it has elimination distance k to the class of empty graphs. So one can think of elimination distance as a natural generalisation of tree-depth.

It is known that the problem of determining the tree-depth of graph is FPT, with tree-depth as the parameter (see [16, Theorem 7.2]). We now give an alternative proof of this, using Theorem 7. It is clear that for any k, the class of graphs of tree-depth at most k is nowhere dense. We show below that it is also first-order definable.

▶ **Proposition 11.** *For each $k \in \mathbb{N}$ there is a first-order formula φ_k such that a graph G has tree-depth k if and only if $G \models \varphi_k$.*

Proof. We use the fact that in a graph of tree-depth less than k, there are no paths of length greater than 2^k. This allows us, in the inductive definition of tree-depth above, to replace the condition of connectedness (which is not first-order definable) with a first-order definable condition on vertices at distance at most 2^k.

Let $\mathrm{dist}_d(u, v)$ denote the first-order formula with free variables u and v that is satisfied by a pair of vertices in a graph G if, and only if, they have distance at most d in G. Note that the formula $\mathrm{dist}_d^{[x.x \neq w]}(u, v)$ is then a formula with three free variables u, v, w which defines those u, v which have a path of length d in the graph obtained by deleting the vertex w.

We can now define the formula φ_k by induction. Only the empty graph has tree-depth 0, so $\varphi_0 := \neg\exists v(v = v)$.

Suppose that φ_k defines the graphs of tree-depth at most k, let

$$\theta_k := (\forall u, v \mathrm{dist}_{2^{k+1}}(u, v)) \wedge \exists w(\varphi_k^{[x.x \neq w]}).$$

The formula θ_k defines the connected graphs of tree depth at most $k + 1$. Indeed, the first conjunct ensures that the graph is connected as no pair of vertices has distance greater than 2^{k+1} and that we can find a vertex w whose removal yields a graph of tree-depth at most k.

We can now define the formula φ_{k+1} as follows.

$$\varphi_{k+1} := (\forall u, v \mathrm{dist}_{2^{k+1}+1}(u, v) \rightarrow \mathrm{dist}_{2^{k+1}}(u, v)) \wedge \forall w \theta_k^{[x.\mathrm{dist}_{2^{k+1}}(w,x)]}.$$

The formula asserts that there are no pairs of vertices whose distance is strictly greater than 2^{k+1} and that for every vertex w, the formula θ_k holds in its connected component, namely those vertices which are at distance at most 2^{k+1} from w. ◀

While the proof of Proposition 9 shows that deletion distance to any slicewise first-order definable class is also slicewise first-order definable, Proposition 11 shows that elimination distance to the particular class of empty graphs is slicewise first-order definable. It does not establish this more generally for elimination distance to any slicewise nowhere dense class.

4 Elimination distance to classes characterised by excluded minors

In this section we show that determining the elimination distance of a graph to a minor-closed class $\mathcal{C}$ is FPT when parameterized by the elimination distance. More generally, we formulate the following parameterized problem where the forbidden minors of $\mathcal{C}$ are also part of the parameter.

Elimination Distance to Excluded Minors
Input: A graph G, a natural number $k \in \mathbb{N}$ and a set of graphs M
Parameter: $k + \sum_{H \in M} |H|$
Problem: Does G have elimination distance k to the class $\mathrm{Forb}(M)$?

It is not difficult to show that the class of graphs which have elimination distance k to a minor-closed class $\mathcal{C}$ is also minor-closed. Indeed, this can be seen directly from an alternative characterisation of elimination distance that we establish below. The characterisation is in terms of the iterated closure of $\mathcal{C}$ under the operation of disjoint unions and taking the class of apex graphs. We introduce a piece of notation for this in the next definition. Recall that we write $\mathcal{C}^{\text{apex}}$ for the class of all the apex graphs over $\mathcal{C}$, and that we write $\overline{\mathcal{C}}$ for the closure of $\mathcal{C}$ under disjoint unions.

▶ **Definition 12.** For a class of graphs $\mathcal{C}$, let $\mathcal{C}_0 := \mathcal{C}$, and $\mathcal{C}_{i+1} := \overline{\mathcal{C}_i^{\text{apex}}}$.

We show next that the class $\mathcal{C}_k$ is exactly the class of graphs at elimination distance k from $\mathcal{C}$.

▶ **Proposition 13.** *Let $\mathcal{C}$ be a class of graphs and $k \geq 0$. Then $\mathcal{C}_k$ is the class of all graphs with elimination distance at most k to $\mathcal{C}$.*

Proof. We prove this by induction. Only the graphs in $\mathcal{C}$ have elimination distance 0 to $\mathcal{C}$, so the statement holds for $k = 0$.

Suppose the statement holds for k. If $G \in \mathcal{C}_{k+1}$, then G is a disjoint union of graphs $G_1, \ldots, G_s$ from ${\mathcal{C}_k}^{\text{apex}}$, so we can remove at most one vertex from each of the G_i and obtain a graph in $\mathcal{C}_k$. Thus the elimination distance of G to $\mathcal{C}_k$ is 1, and by induction the elimination distance to $\mathcal{C}$ is $k + 1$. Conversely, if G has elimination distance $k + 1$ to $\mathcal{C}$, then we can remove a vertex from each component of G to obtain a graph G' with elimination distance k to $\mathcal{C}$. Using the induction hypothesis each component of G' is in $\mathcal{C}_k$, and thus $G \in \mathcal{C}_{k+1}$. ◀

It is easy to see that if $\mathcal{C}$ is a minor-closed class of graphs then so is $\mathcal{C}_k$ for any k. Indeed, it is well-known that C^{apex} is minor-closed for any minor-closed $\mathcal{C}$, so we just need to note that $\overline{\mathcal{C}}$ is also minor-closed. But it is clear that if H is a minor of a graph G that is the disjoint union of graphs $G_1, \ldots, G_s$, then H itself is the disjoint union of minors of $G_1, \ldots, G_s$. Thus, the class of graphs of elimination distance at most k to a minor-closed class $\mathcal{C}$ is itself minor-closed. We next show that we can construct the set of its minimal excluded minors from the corresponding set for $\mathcal{C}$.

To obtain $M(\mathcal{C}_k)$, we need to iteratively compute $M(\mathcal{C}^{\text{apex}})$ and $M(\overline{\mathcal{C}})$ from $M(\mathcal{C})$. Adler et al. [1] show that from the set of minimal excluded minors $M(\mathcal{C})$ of a class $\mathcal{C}$, we can compute $M(\mathcal{C}^{\text{apex}})$:

▶ **Theorem 14** ([1], Theorem 5.1)**.** *There is a computable function that takes the set of graphs $M(\mathcal{C})$ characterising a minor-closed class $\mathcal{C}$ to the set $M(\mathcal{C}^{apex})$.*

We next aim to show that from $M(\mathcal{C})$ we can also compute $M(\overline{\mathcal{C}})$. Together with Theorem 14 this implies that from $M(\mathcal{C})$ we can compute $M(\mathcal{C}_k)$.

We begin by characterising minor-closed classes that are closed under disjoint unions in terms of the connectedness of their excluded minors.

▶ **Lemma 15.** *Let $\mathcal{C}$ be a class of graphs closed under taking minors. Then $\mathcal{C}$ is closed under taking disjoint unions iff each graph in $M(\mathcal{C})$ is connected.*

Proof. Let $\mathcal{C}$ be a minor-closed class of graphs, and let $M(\mathcal{C})$ be its set of minimal excluded minors.

Suppose each of the graphs in $M(\mathcal{C})$ is connected. Let $H \in M(\mathcal{C})$ and let $G = G_1 \oplus \cdots \oplus G_r$ be the disjoint union of graphs $G_1, \ldots, G_r \in \mathcal{C}$. Because H is connected, we have that $H \preceq G$ if, and only if, $H \preceq G_i$ for some i. So, since for each i, $G_i \in \mathcal{C}$, we have $H \npreceq G$ and thus $G \in \mathcal{C}$. This shows that $\mathcal{C}$ is closed under taking disjoint unions.

Conversely assume $H \in M(\mathcal{C})$ is not connected and let $A_1, \ldots, A_t$ be its connected components. Then $A_1, \ldots, A_t \in \mathcal{C}$, since each A_i is a proper minor of H, and H is minor-minimal in the complement of $\mathcal{C}$. However, $A_1 \oplus \cdots \oplus A_t = H \notin \mathcal{C}$. ◀

▶ **Definition 16.** For a graph G with connected components $G_1, \ldots, G_r$, let $\mathcal{H}$ denote the set of *connected* graphs H with $V(H) = V(G)$ and such that the subgraph of H induced by $V(G_i)$ is exactly G_i. We define the *connection closure* of G to be the set of all minimal (under the subgraph relation) graphs in $\mathcal{H}$. The connection closure of a set of graphs is the union of the connection closures of the graphs in the set.

Note that if G has e edges and m components, then any graph in the connection closure of G has exactly $e + m - 1$ edges. This is because it has G as a subgraph and in addition $m - 1$ edges corresponding to a tree on m vertices connecting the m components.

▶ **Lemma 17.** *Let $\mathcal{C}$ be a minor-closed class of graphs. Then $M(\overline{\mathcal{C}})$ is the set of minor-minimal graphs in the connection closure of $M(\mathcal{C})$.*

Proof. Let $\mathcal{C}$ be a minor-closed class of graphs, with $M(\mathcal{C})$ its set of minimal excluded minors, and let $\hat{M}$ be the connection closure of $M(\mathcal{C})$.

Let G be a graph such that $\hat{H} \npreceq G$ for all $\hat{H} \in \hat{M}$. Suppose for contradiction that G is not a disjoint union of graphs from $\mathcal{C}$. Then there is a component G' of G that is not in $\mathcal{C}$ and therefore there is a graph $H \in M(\mathcal{C})$ such that $H \preceq G'$. We show that one of the graphs in the connection closure of H is a minor of G'.

Let $\{w_1, \ldots, w_s\}$ be the vertex set of H and consider the image $T_1, \ldots, T_s$ of the minor map from H to G'. Let T be a minimal subtree of G' that contains all of the T_i. Such a tree must exist since G' is connected. Let $\hat{H}$ be the graph with the same vertex set as H, and an edge between two vertices w_i, w_j whenever either $w_i w_j \in E(H)$ or when there is a path between T_{w_i} and T_{w_j} in T that is disjoint from any T_{w_k} with $w_i \neq w_k \neq w_j$. We claim that $\hat{H}$ is in the connection closure of H. By construction, $\hat{H}$ is connected and contains all components of H as disjoint subgraphs, so we only need to argue minimality. $\hat{H}$ has no vertices besides those in H so no graph obtained by deleting a vertex would contain all components of H as subgraphs. To see that no edge of $\hat{H}$ is superfluous, we note it has exactly $e + m - 1$ edges and thus no proper subgraph could be connected and have all components of H as disjoint subgraphs. By the construction $\hat{H} \preceq G' \preceq G$, so by the transitivity of the minor relation we have that $\hat{H} \preceq G$.

Conversely let G be an arbitrary graph and assume that $\hat{H} \in \hat{M}$ and $\hat{H} \preceq G$. Because $\hat{H}$ is connected, there is a connected component G' of G such that $\hat{H} \preceq G'$. Now there must be a graph $H \in M(\mathcal{C})$ such that $\hat{H}$ is in the connection closure of H, and since H is a subgraph of $\hat{H}$, $H \preceq \hat{H}$. Then, by the transitivity of the minor relation, $H \preceq G'$ and thus $G' \notin \mathcal{C}$. Therefore G is not a disjoint union of graphs from $\mathcal{C}$. ◀

Now our main theorem is established by a simple induction:

▶ **Theorem 18.** *There is a computable function which takes a set M of excluded minors characterising a minor-closed class $\mathcal{C}$ and $k \geq 0$ to the set $M(\mathcal{C}_k)$.*

Proof. The proof is by induction. For $k = 0$, the set of minimal excluded minors of $\mathcal{C}_0$ is $M(\mathcal{C}_0) = M(\mathcal{C})$, which is given. For $k > 0$, we have that $\mathcal{C}_k = \overline{\mathcal{C}_{k-1}{}^{\text{apex}}}$. By the induction hypothesis we can compute $M(C_{k-1})$, by Theorem 14 we can compute $M(\mathcal{C}_{k-1}{}^{\text{apex}})$ and using Lemma 17 we can compute the connection closure of $M(\mathcal{C}_{k-1}{}^{\text{apex}})$ to obtain $M(\overline{\mathcal{C}_{k-1}{}^{\text{apex}}}) = M(\mathcal{C}_k)$. ◀

So by the Robertson-Seymour Theorem we have the following:

▶ **Corollary 19.** *Let $\mathcal{C}$ be a minor-closed graph class. Then the problem* Elimination Distance to Excluded Minors *is* FPT.

5 Conclusion

We are motivated by the study of the fixed-parameter tractability of edit distances in graphs. Specifically, we are interested in edit distances such as the number of vertex or edge deletions, as well as more involved measures like elimination distance. Aiming at studying general techniques for establishing tractability, we establish an algorithmic meta-theorem showing that any slicewise first-order definable and slicewise nowhere dense problem is FPT. This yields, for instance, the tractability of counting the number of vertex and edge deletions to a class of bounded degree. As a second result, we establish that determining elimination distance to any minor-closed class is FPT, answering an open question of [2].

A natural open question raised by these two results is whether elimination distance to the class of graphs of degree d is FPT. When d is 0, this is just the tree-depth of a graph, and this case is covered by our first result. For positive values of d, it is not clear whether elimination distance is first-order definable. Indeed, a more general version of the question is whether for any nowhere dense and first-order definable $\mathcal{C}$, elimination distance to $\mathcal{C}$ is FPT.

Another interesting case that seems closely related to our methods, but is not an immediate consequence is that of classes that are given by first-order interpretations from nowhere dense classes of graphs. For instance, consider the problem of determining the deletion distance of a graph to a disjoint union of complete graphs. This problem, known as the cluster vertex deletion problem is known to be FPT (see [14]). The class of graphs that are disjoint unions of cliques is first-order definable but certainly not nowhere dense and so the method of Section 3 does not directly apply. However, this class is easily shown to be interpretable in the nowhere dense class of forests of height 1. Can this fact be used to adapt the methods of Section 3 to this class?

References

1 I. Adler, M. Grohe, and S. Kreutzer. Computing excluded minors. In *SODA'08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, January 2008.

2 J. Bulian and A. Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. In *Parameterized and Exact Computation – 9th International Symposium, IPEC 2014, Wroclaw, Poland, September 10–12, 2014. Revised Selected Papers*, pages 135–146, 2014.

3 L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58:171–176, 1996.

4 R. Diestel. *Graph Theory*. Springer, January 2000.

5 R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, October 2012.

6 J. Flum and M. Grohe. Fixed-Parameter Tractability, Definability, and Model-Checking. *SIAM J. Comput.*, 31(1):113–145, 2001.
7 J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, May 2006.
8 F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh. Planar F-deletion: Approximation, kernelization and optimal FPT algorithms. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 470–479, 2012.
9 J. Gajarský, P. Hlinený, J. Obdrzálek, S. Ordyniak, F. Reidl, P. Rossmanith, F. Sanchez Villaamil, and S. Sikdar. Kernelization using structural parameters on sparse graph classes. In *Algorithms – ESA 2013 – 21st Annual European Symposium*, pages 529–540, 2013.
10 P. A. Golovach. Editing to a Graph of Given Degrees. In *Parameterized and Exact Computation*, pages 196–207. Springer, September 2014.
11 M. Grohe, S. Kreutzer, and S. Siebertz. Deciding first-order properties of nowhere dense graphs. In *STOC'14: Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. ACM, May 2014.
12 J. Guo, F. Hüffner, and R. Niedermeier. A Structural View on Parameterizing Problems: Distance from Triviality. In *Parameterized and Exact Computation*, pages 162–173. Springer, 2004.
13 W. Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997.
14 F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier. Fixed-parameter algorithms for cluster vertex deletion. *Theory Comput. Syst.*, 47:196–217, 2010.
15 D. Marx. Parameterized coloring problems on chordal graphs. *Theor. Comput. Sci.*, 351:407–424, 2006.
16 J. Nesetril and P. Ossona de Mendez. *Sparsity – Graphs, Structures, and Algorithms*. Springer, 2012.
17 R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, February 2006.
18 N. Robertson and P. D. Seymour. Graph minors. XX. wagner's conjecture. *J. Comb. Theory, Ser. B*, 92:325–357, 2004.