

Fixed-Parameter Tractable Distances to Sparse Graph Classes

Jannis Bulian¹ · Anuj Dawar¹

Received: 11 January 2016 / Accepted: 18 October 2016 / Published online: 31 October 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

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. We demonstrate that several prior results (of Golovach, Moser and Thilikos and Mathieson) on the fixed-parameter tractability of distance measures are special cases of our first method.

Keywords Fixed-parameter tractable · Parameterized complexity · Graph theory · Sparse graphs · Nowhere dense · Excluded minor · Minor-closed · Deletion distance · Elimination distance · Distance

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

✉ Jannis Bulian
jannis.bulian@cl.cam.ac.uk

¹ Computer Laboratory, University of Cambridge, Cambridge, UK

al. [19]). Other examples of this include the study of modulators to graphs of bounded tree-width in the context of kernelization (see [13, 15]) or the parameterizations of colouring problems (see [22]). 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 [4] 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, as argued in [19]. Another recently introduced measure is *elimination distance*, defined in [3] 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 techniques for establishing that such a distance measure is FPT. The first builds on the recent result of Grohe et al. [18] 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 degree-constrained graphs are FPT. We illustrate the power of this method by showing that it includes as special cases prior results by Golovach [16], Moser and Thilikos [26] and Mathieson [23, 24] obtained by more specific 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} may not, in general, be 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 [27, 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 should 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 [3]. 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 Sect. 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 Sect. 4 gives some applications of the general method. Section 5 establishes that the problem of determining elimination distance to any minor-closed class is FPT. Some open questions are discussed in Sect. 6.

2 Preliminaries

First-Order Logic We assume some familiarity with first-order logic for Sect. 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 will mostly be (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:

$$\phi := R(x_1, \dots, x_k) \mid x = y \mid \neg\phi \mid \phi \vee \psi \mid \exists x.\phi,$$

where k is the arity of the relation symbol R .

We also use the following abbreviations:

$$\phi \wedge \psi := \neg(\neg\phi \vee \neg\psi), \quad \forall x.\phi := \neg\exists x.\neg\phi.$$

The *quantifier rank* of a formula ϕ is the nesting depth of quantifiers in ϕ . For a more detailed presentation we refer to Hodges [20].

We sometimes need to define formulas of first-order logic by *relativisation*, and we define the notion here.

Definition 1 Let ϕ and $\psi(x)$ be first-order formulas, where ψ has a distinguished free variable x which does not appear in ϕ . The *relativisation* of ϕ by $\psi(x)$, denoted $\phi^{[x.\psi]}$ is defined recursively by the rules:

- for atomic ϕ , $\phi^{[x.\psi]}$ is the same as ϕ ;
- $(\phi_1 \vee \phi_2)^{[x.\psi]}$ is $\phi_1^{[x.\psi]} \vee \phi_2^{[x.\psi]}$;
- $(\neg\phi)^{[x.\psi]}$ is $\neg(\phi)^{[x.\psi]}$; and

- $(\exists v \varphi')^{[x.\psi]}$ is $\exists v(\psi[v/x] \wedge (\varphi')^{[x.\psi]})$. 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 a graph G if, and only if, φ is true in the subgraph of G induced by the vertices that satisfy $\psi(x)$. In particular the relativisation of $\forall v \varphi'$ is $\forall v(\psi[v/x] \rightarrow (\varphi')^{[x.\psi]})$.

Note that the variable x that is free in ψ is bound in $\varphi^{[x.\psi]}$. Other variables that appear free in ψ remain free in $\varphi^{[x.\psi]}$. We stress this as it is needed in Proposition 2 where nested relativisations are used.

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, following the textbook of Flum and Grohe [12]. A *problem* $Q \subseteq \text{str}(\sigma)$ is an (isomorphism-closed) class of σ -structures given some signature σ . A *parameterization* is a computable function $\kappa : \text{str}(\sigma) \rightarrow \mathbb{N}$. We say that Q is *fixed-parameter tractable* with respect to κ if we can decide whether an input $A \in \text{str}(\sigma)$ 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 [9], Flum and Grohe [12] and Niedermeier [29].

A parameterized problem (Q, κ) is *slice-wise first-order definable* if there is a computable function $f : \mathbb{N} \rightarrow \text{FO}[\sigma]$ such that a σ -structure A with $\kappa(A) \leq i$ is in Q if, and only if, $A \models f(i)$. Slice-wise definability of problems in a logic was introduced by Flum and Grohe [11].

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 [8]. 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. $T_u \cap T_v = \emptyset$, 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 $H \preceq G$ and $G \in \mathcal{C}$ 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 [30] 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 $\text{Forb}(M)$ for the class of graphs which forbid M as minors, i.e. $\text{Forb}(M) = \{G \mid H \not\preceq G \text{ for all } H \in M\}$.

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 h from integers to graphs such that if $G \in \mathcal{C}$ then for all r we have $h(r) \not\preceq_r G$. 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 if $G \in Q$ and $\kappa(G) \leq i$ then for all r we have $h(i, r) \not\preceq_r G$. We 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 $\bar{\mathcal{C}}$. 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 $v_1, \dots, v_k \in V(G)$ such that $G \setminus \{v_1, \dots, v_k\} \in \mathcal{C}$.

A richer notion of distance to a graph class \mathcal{C} , inspired by the definition of tree-depth, is introduced in [3], and we define it next.

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 Sect. 4 we show that this provides a unifying account of a number of existing results in the literature by giving a single method of proof for them.

The method is an adaptation of the main algorithm in Grohe et al. [18]. Since the proof of our results is essentially a modification of the central construction in [18], 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 [18]. In Sect. 3.1 we give an overview of the key elements of the construction from [18] and state all the definitions we require to formulate our

results. Section 3.2 then gives our main result and Sect. 4 derives some consequences for distance measures.

3.1 Evaluating Formulas on Nowhere Dense Classes

The key result of [18] is:

Theorem 1 [18, 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} .*

The proof of this theorem rests, in turn, on two others, stated as Theorems 2 and 3 below. In order to formally state those results and explain how they can be adapted to our purposes, we need to formulate some definitions.

The algorithm developed in the proof of Theorem 1 uses a locality-based approach, similar to that used by Frick and Grohe [14] to show that first-order evaluation is FPT on graphs of locally bounded treewidth and developed in [7] for application to graph classes with locally excluded minors. The key to these is an application of Gaifman's locality theorem [10, Thm. 2.5.1]. Let $\text{dist}_d(u, v)$ 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 . A *basic local sentence* of radius d is a first-order formula of the form:

$$\exists x_1 \cdots \exists x_k \bigwedge_{1 \leq i < j \leq k} \neg \text{dist}_{2d}(x_i, x_j) \wedge \bigwedge_{1 \leq i \leq k} \theta^{[x.\text{dist}_d(x, x_i)]}, \quad (1)$$

where θ is some first-order sentence. Essentially, this basic local sentence asserts the existence of k disjoint neighbourhoods of radius r , each of which satisfies the sentence θ . Gaifman's locality theorem then asserts that every first-order sentence φ is equivalent to a Boolean combination of basic local sentences. Moreover, this equivalent *Gaifman normal form* can be computed from φ . Thus, in considering the parameterized complexity of evaluating φ in a graph G it suffices to consider just basic local sentences.

To evaluate a basic local sentence such as (1) in a graph G , it suffices to determine for each vertex v whether the neighbourhood of radius d around it satisfies θ and then check whether in set of vertices for which this is true there is a $2d$ -independent set. If the neighbourhoods in G are structurally simpler than G , this may provide an efficient means of evaluating a first-order sentence. For instance, in a class \mathcal{C} of bounded local treewidth, the d -neighbourhoods of vertices have bounded treewidth, and if \mathcal{C} has locally excluded minors then for every d , the d -neighbourhoods exclude some graph as a minor. If \mathcal{C} is a nowhere-dense class, we do not have a structurally simpler characterisation of the d -neighbourhoods occurring in graphs in \mathcal{C} . Instead we rely on the property that such classes are *quasi-wide*. This is a notion of sparseness for graphs introduced in [5, 6] (for the connection to nowhere-dense classes, see [27]). A graph class \mathcal{C} is said to be quasi-wide if there is a function s such that for any d and N and any graph $G \in \mathcal{C}$, if A is a sufficiently large set of more vertices in G , then

G contains a *bottleneck* set S of at most $s(d)$ vertices such that in $G \setminus S$, A contains a d -independent set of size N .

This suggests strategy for evaluating a formula φ in a graph G that comes from a class \mathcal{C} that is nowhere dense and closed under taking subgraphs. We identify a bottleneck set S and remove these vertices from the graph, colouring each remaining vertex v according to which elements of S are neighbours of v . It is easy to translate the formula φ into a formula φ' in a vocabulary expanded with these colours such that φ' is true in the coloured graph $G \setminus S$ if, and only if, φ is true in G . The existence of a large set of vertices in $G \setminus S$ that are pairwise far apart means that we can evaluate local sentences in neighbourhoods around these vertices in (total) amortized time dependent on the size of the graph. This evaluation is done recursively, since the neighbourhood of a vertex v is also in the class \mathcal{C} (by the assumption of closure under taking subgraphs). The difficulty with this approach is that, if all we know about the d -neighbourhood of a vertex is that it is also in \mathcal{C} , we cannot bound the depth of recursion by a constant. Thus the vocabulary of the formulas φ' we construct, and hence also their size, is no longer dependent solely on parameters. Grohe et al. circumvent this difficulty through two innovative methods. The first is to define a colouring based on *sparse neighbourhood covers* and show that such covers exist in nowhere-dense classes of graphs. The second is an alternative way of amortizing the quantifier rank of the recursively defined formulas in the expanded vocabularies, defining a *discounted* measure of rank. We expand on these notions next.

For $r \in \mathbb{N}$, an r -neighbourhood cover of a graph G is a set \mathcal{X} of connected subgraphs of G such that for every $v \in V(G)$ there is an $X \in \mathcal{X}$ that contains the r -neighbourhood of v . The elements of \mathcal{X} are called *clusters*. The *radius* of \mathcal{X} is the maximum radius of any of its clusters. The *degree* of a vertex v is the number of clusters that contain v and the *maximum degree* of \mathcal{X} is the maximum over all $v \in V(G)$ of the degree of v . These definitions allow us to state the following theorem.

Theorem 2 [18, 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.*

In this theorem, f is a function of r and ϵ and depends on the class \mathcal{C} in the sense that it is determined, for an effectively nowhere dense \mathcal{C} by its parameter function. While the algorithm of [18] 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 of the proof of Theorem 2.

Lemma 1 *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.*

The logic FO^+ is defined by extending FO with an atomic formula $\text{dist}_d(x, y)$ for each $d \in \mathbb{N}$ and each pair of variables x, y . The idea is that in this expanded logic, the assertion that x and y are at distance at least d does not require any quantifiers. However, we lose a key feature of first-order logic that is essential to parameterized algorithms and that is that, for each q , there are, up to equivalence, only finitely many distinct sentences of FO of quantifier-rank q . To recover this property for FO^+ , Grohe et al. define a discounted quantifier rank measure as follows. We say that φ of FO^+ has q -rank m if it has quantifier-rank at most m and for any atomic subformula $\text{dist}_d(x, y)$ which occurs within the scope of i quantifiers we have $d \leq (4q)^{q+m-i}$. It can then be shown that there are for each k, q and m , up to equivalence, only finitely many distinct formulas of FO^+ with q -rank m in the free variables x_1, \dots, x_k . Fix $\varphi^+(\sigma, k, q, m)$ a finite set of representative formulas in the vocabulary σ , including one of each class up to logical equivalence. We then define the vocabulary $\sigma \star q$ as the expansion of σ with a unary relation P_φ for each $\varphi \in \varphi^+(\sigma, 1, q, q)$.

Given a σ -structure A and \mathcal{X} an r -neighbourhood cover of A , fix for each $v \in V(A)$ a neighbourhood $\mathcal{X}(v) \in \mathcal{X}$ such that the r -neighbourhood of v is contained in $\mathcal{X}(v)$. Let $A \star_{\mathcal{X}} q$ be the $\sigma \star q$ expansion of A in which P_φ is interpreted by the set of v in which the substructure of A induced by $\mathcal{X}(v)$ satisfies $\varphi[v]$. We now inductively define the vocabularies $\sigma \star^i q$ by: $\sigma \star^0 q$ is σ and $\sigma \star^{i+1} q$ is $(\sigma \star^i q) \star q$. Similarly, we define the structure $A \star_{\mathcal{X}}^0 q$ to be A and $A \star_{\mathcal{X}}^{i+1} q$ to be $(A \star_{\mathcal{X}}^i q) \star q$. A (q, r) -independence sentence is a sentence of FO^+ of the form:

$$\exists x_1 \dots \exists x_k \bigwedge_{1 \leq i < j \leq k} \neg \text{dist}_{2r}(x_i, x_j) \wedge \bigwedge_{1 \leq i \leq k} \theta(x_i),$$

where θ is quantifier-free. Note that such a sentence is a basic local sentence (as in (1)) with the stronger requirement that θ is quantifier-free. We are now ready to state the Rank-Preserving Locality Theorem.

Theorem 3 (Rank-preserving locality theorem [18, 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 an FO^+ -formula $\hat{\varphi}(x)$, which is a Boolean combination of $(q + 1, r)$ -independence sentences and atomic formulas, such that for any graph G every r -neighbourhood cover \mathcal{X} of G , and every $v \in V(G)$,*

$$G \models \varphi(v) \iff G \star_{\mathcal{X}}^{q+1} q \models \hat{\varphi}(v).$$

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

An important tool for constructing $G \star_{\mathcal{X}}^{q+1} q$ is a game characterisation of nowhere dense classes. The game has three parameters: ℓ, m, r . In the (ℓ, m, r) Splitter game two players Connector and Splitter play against each other. In each round Connector chooses a vertex u , and Splitter has to respond with a set A of vertices of size at most m in the r -neighbourhood of u . In the next round the graph is the neighbourhood of u with the vertices from A removed. If the graph is empty, Splitter wins. If Connector survives for more than ℓ rounds, she wins. Grohe et al. [18, Theorem 4.2] prove that if

\mathcal{C} is a nowhere dense class, then there are ℓ, m such that Splitter has a winning strategy on the $(\ell, m, 2r)$ Splitter game on every graph in \mathcal{C} .

The Splitter’s strategy on a graph G (which can be efficiently computed) is the essential tool in the construction of $G \star_{\mathcal{X}}^{q+1} q$. The inductive procedure used to compute $G \star_{\mathcal{X}}^{q+1} q$ from G is outlined in [18, Proof of Theorem 8.1]. We note that the termination of the algorithm depends on the length of the game—which is bounded by a constant since \mathcal{C} is nowhere dense. The strategy to compute Splitter’s moves is described in [18, Remark 4.3]. Since the run time of the algorithm to compute $G \star_{\mathcal{X}}^{q+1} q$ only depends on q and the length of the Splitter game and we can compute this in advance, we can once again extract the fact that if we start with an arbitrary graph G , we can efficiently either transform it into $G \star_{\mathcal{X}}^{q+1} q$ or determine that it is not in the class \mathcal{C} . This is summed up in the following lemma.

Lemma 2 *Let \mathcal{C} be a nowhere dense class of graphs. For every $\epsilon > 0$ there is an algorithm that runs in time $O(f(q) \cdot n^{1+\epsilon})$ for some function f , and which given a graph G returns $G \star_{\mathcal{X}}^{q+1} q$ or determines that $G \notin \mathcal{C}$.*

Theorem 3 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:

<p>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?</p>
--

The problem is shown to be FPT on nowhere dense classes of graphs [18, Theorem 5.1]. Since the runtime of the algorithm depends on the length of the Splitter game and Splitter’s strategy, and this can be bounded in advance, [18, Theorem 5.1] can be restated as follows:

Lemma 3 *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 that $G \notin \mathcal{C}$. Furthermore, if \mathcal{C} is effectively nowhere dense, then f is computable.*

This is all we need to evaluate $\hat{\phi}$ on $G \star_{\mathcal{X}}^{q+1} q$, which is equivalent to evaluating ϕ on G by Theorem 3.

3.2 Deciding Definable Nowhere Dense Problems

The main result of [18] establishes that checking whether $G \models \phi$ 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. In Sect. 3.1

above we give 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 $G \models \phi$ 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 4 *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 parameter 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 3), 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 1, 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 2 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 3. Since evaluating $\hat{\varphi}$ on G' is equivalent to evaluating φ on G this allows us to decide whether $G \in Q$.

□

4 Applications

In this section we illustrate the power of Theorem 4 by showing that a number of known fixed-parameter tractability results can be obtained as direct consequences of the theorem. In Sect. 4.1, we start with the simple observation that if a parameterized problem is slicewise first-order definable then so is deletion distance to the problem (suitably parameterized). Examples of this include previous results of Moser and Thilikos [26]. We consider more general edit distances in Sect. 4.2 and show that of Mathieson [23, 25] can be obtained as special cases of our result. Finally, in Sect. 4.3, we consider the problem of computing the tree-depth of a graph. Again, this problem is known to be FPT, and the novelty here is in constructing the first-order definitions that show it is slicewise first-order definable. The consideration of tree-depth also leads naturally to considering, more generally, elimination distance to sparse classes and this topic is taken up in Sect. 5.

4.1 Deletion Distance

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:

<p>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 $G \setminus S \in Q$?</p>

Proposition 1 *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:

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

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

To see that DELETION DISTANCE TO Q is also slicewise nowhere dense, let h be the parameter function for Q . If the graph $h(i, r)$ has m vertices, then K_m is not a depth- r -minor of any graph in \mathcal{C}_i . Then a graph which has deletion distance k to \mathcal{C}_i cannot have K_{m+k} as a depth- r -minor. Indeed, suppose $K_{m+k} \preceq_r G$ and $G \setminus S \in \mathcal{C}_i$ where S is a set of k vertices. Vertices from S can appear in the images of at most k vertices from K_{m+k} under the minor map. Thus, this minor map also witnesses that $K_m \preceq_r G \setminus S$, a contradiction. \square

Consider as an example deletion distance k to maximum degree d .

Example 1 We can express that a vertex has no more than d different neighbours using the following formula:

$$\varphi_d(v) := \neg \exists v_1, \dots, v_{d+1} \left(\left(\bigwedge_{i < j} v_i \neq v_j \right) \wedge \left(\bigwedge_i E(v, v_i) \right) \right).$$

So a graph G has maximum degree bounded by d if and only if $G \models \forall v \varphi_d(v)$. Let \mathcal{C}_d be the class of graphs with maximum degree bounded by d . Then the following formula captures deletion distance k to maximum degree d :

$$\exists w_1, \dots, w_k \forall v \left(\bigwedge_i v \neq w_i \right) \rightarrow \left(\neg \exists v_1, \dots, v_{d+1} \left(\left(\bigwedge_{i,j} v_i \neq w_j \right) \wedge \left(\bigwedge_{i < j} v_i \neq v_j \right) \wedge \left(\bigwedge_i E(v, v_i) \right) \right) \right).$$

Thus the technique introduced in this section is sufficient to show that deletion distance k to maximum degree d is fixed-parameter tractable parameterized by $k + d$.

Moser and Thilikos [26] showed that deleting k vertices to obtain a d -regular graph is fixed-parameter tractable parameterized by $k + d$. Since the class of d -regular graphs is also first-order definable and nowhere dense for any d , their result is also a consequence of Theorem 4.

4.2 Edit Distances to Graph Classes Defined by Degree Constraints

Instead of deletion distance (defined by deleting vertices), we can also consider more general graph editing distances (defined through more general edit operations on the graph), e.g. modifying the graph by adding or deleting edges.

For example, to obtain a formula that defines the graphs that are one edge addition away from a class of graphs \mathcal{C} defined by the formula φ , we construct $\hat{\varphi}$ from φ by replacing all occurrences of $E(w_1, w_2)$ in φ by:

$$(w_1 = u \wedge w_2 = v) \vee (w_1 = v \wedge w_2 = u) \vee E(w_1, w_2).$$

Then the formula $\exists u \exists v \hat{\varphi}$ defines the class of graphs with edge addition distance 1 to \mathcal{C} , i.e. $G \models \exists u \exists v \hat{\varphi}$ if, and only if, G with an additional edge satisfies φ . This can easily be generalised to k edge additions: From a formula φ we can obtain a formula $\hat{\varphi}_k$ such that for any graph G we have that $G \models \hat{\varphi}_k$ if, and only if, there are pairs of vertices $u_1, v_1, \dots, u_k, v_k \in V(G)$ such that G , with additional edges $u_1 v_1, \dots, u_k v_k$, satisfies φ .

Similarly, given a formula ψ that defines a graph class \mathcal{C} , we can define a formula $\hat{\psi}$ by replacing all occurrences of $E(w_1, w_2)$ in ψ by:

$$(w_1 \neq u \vee w_2 \neq v) \wedge (w_1 \neq v \vee w_2 \neq u) \wedge E(w_1, w_2).$$

Then $\exists u \exists v \hat{\psi}$ defines the class of graphs with edge deletion distance 1 to \mathcal{C} . It should be clear that this can also be generalised to k edge deletions.

Thus, an analogue of Proposition 1 can be obtained for any edit distance where the allowed edit operations are a combination of vertex and edge deletions and additions. In the following we discuss this in more detail, where the class we are editing to is defined by degree constraints.

In his doctoral thesis Mathieson [23] studies the parameterized complexity of such graph editing problems with the aim to satisfy a variety of degree constraints. He defines the general template of WEIGHTED DEGREE CONSTRAINED EDITING (or just

WDCE). In the following we explore one instance (that he refers to as WDCE'_1) of a number of more general templates that also allow for weight functions of vertices and edges, as well as a different degree target for each vertex and a target for the sum of edge weights. This is just for the sake of simplicity here, the more general operations are also definable in first-order logic and give rise to nowhere dense graph classes. In the following we abbreviate the editing operations vertex deletion, edge deletion and edge addition as v , e and a respectively. Then for each non-empty $S \subseteq \{v, e, a\}$ define $\text{WDCE}(S)$ as:

WEIGHTED DEGREE CONSTRAINED EDITING(S) ($\text{WDCE}(S)$)

Input: A graph G , two integers k and d

Parameter: $k + d$

Problem: Can we obtain from G a graph G' using k editing operations from S only, such that all vertices of G' have degree d ?

Mathieson [25] shows that the problem is fixed-parameter tractable for any S and parameter $k + d$. Inspired by Stewart [31], Mathieson shows that the problem is first-order definable (with the size of the formula depending on k and d), by considering the incidence graph as a relational structure. (For the weighted version of the problem, he adds a unary relation for every possible weight.) Since a graph that can be edited to be regular must its degree bounded by $k + d$ it is therefore also nowhere dense. Thus the result also follows directly from Theorem 4.

Building on this, Golovach [16] gives a concrete algorithm that edits a graph so that every vertex has a given degree at most d using at most k edge additions/deletions.

More recently, Mathieson [24] looks at more general versions of degree constraint problems. He considers three notions of regularity: *edge-degree-regular*, *edge-regular* and *strongly-regular*. He studies the problems of editing to these three notions of regularity.

The *edge-degree* of an edge uv is the sum of the degrees of the endpoints of $d(u) + d(v)$ and a graph is *edge-degree-regular* if all edges uv have the same edge-degree.

The two other notions combine the degrees of vertices and common neighbourhoods of endpoints of edges (and non-edges). A graph is (r, λ) -*edge-regular* if every vertex has degree r and every edge uv has $|N(u) \cap N(v)| = \lambda$. A graph is (r, λ, μ) -*strongly-regular* if it is (r, λ) -edge-regular and for every pair u, v of non-adjacent vertices we have $|N(u) \cap N(v)| = \mu$. This is the standard notion of a strongly regular graph as introduced by Bose [2].

Just as above we abbreviate the editing operations vertex deletion, edge deletion and edge addition as v , e and a respectively. We also abbreviate the three notions of regularity edge-degree regular, edge-regular and strongly regular as r_1, r_2, r_3 respectively. Then for each non-empty $S \subseteq \{v, e, a\}$ and each $r \in \{r_1, r_2, r_3\}$ define $\text{RCE}(S)$ as:

Mathieson [24] shows that editing to these three notions of regularity is FPT parameterized by $k + d$, where d is the degree we are editing to and k is the number of allowed edits. This also follows from our meta-theorem: Just as in Example 1, it is easy to show

REGULARITY CONSTRAINED EDITING (S, r) (RCE (S, r))

Input: A graph G , integers k, d (and additionally an integer λ if $r = r_2$ or r_3 , and an integer μ if $r = r_3$)

Parameter: $k + d$

Problem: Can we obtain from G a graph G' using editing operations from S only, such that G' is r -regular (with the given parameters)?

that the class of graphs with edit distance k to an d -regular graph is first-order definable and has bounded degree, and is thus also nowhere dense. The additional constraints are also first-order definable, for example $|N(u) \cap N(v)| = \lambda$ can be expressed as follows:

$$\exists x_1 \dots \exists x_\lambda \cdot \left(\bigwedge_{i < j} x_i \neq x_j \wedge \bigwedge_i (E(x_i, u) \wedge E(x_i, v)) \wedge \neg \exists y \left(\bigwedge_i (x_i \neq y) \wedge (E(u, y) \wedge E(v, y)) \right) \right)$$

So each of these problems is first-order definable and nowhere-dense, so it follows directly from Theorem 4 that these problems are fixed-parameter tractable with the combined parameter $k + d$.

4.3 Tree-Depth

Recall that 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, and can itself be interpreted as a distance measure (elimination distance to the empty graph). For convenience we give the usual definition here:

Definition 2 The *tree-depth* of a graph G , written $td(G)$, is defined as follows:

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

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 [28, Theorem 7.2]). We now give an alternative proof of this, using Theorem 4. 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 2 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 [28, Section 6.2]. 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 .

Recall that $\text{dist}_d(u, v)$ is 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 $\text{dist}_d^{[x.x \neq w]}(u, v)$ 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 $\phi_0 := \neg \exists v (v = v)$.

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

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

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 the second conjunct ensures 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.

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

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 1 shows that deletion distance to any slicewise first-order definable class is also slicewise first-order definable, Proposition 2 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—that remains an open question. We conjecture that elimination distance to a slicewise nowhere dense class is not first-order definable.

5 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: $k + \sum_{G \in M} |G|$

Problem: Does G have elimination distance k to the class $\text{Forb}(M)$?

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 a minor-closed class. 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 3 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 3 *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, \dots, 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 $G \in \mathcal{C}_{k+1}$. \square

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 $\mathcal{C}^{\text{apex}}$ 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, \dots, G_s , then H itself is the disjoint union of minors of G_1, \dots, 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 5 [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 $M(\mathcal{C}^{\text{apex}})$.*

We next aim to show that from $M(\mathcal{C})$ we can also compute $M(\overline{\mathcal{C}})$. Together with Theorem 5 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 4 *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 $M(\mathcal{C}) = \{H_1, \dots, H_s\}$ 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 $G = G_1 \oplus \dots \oplus G_r$ be the disjoint union of graphs $G_1, \dots, G_r \in \mathcal{C}$. Because H is connected, we have that $H \preceq G$ if and only if $H \preceq G_i$ for one $1 \leq i \leq r$. So, since all the $G_i \in \mathcal{C}$, we have $H \not\preceq G$ and thus $G \in \mathcal{C}$. This shows that \mathcal{C} is closed under taking disjoint unions.

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

Definition 4 For a graph G with connected components G_1, \dots, 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 5 *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 $\hat{H} \not\preceq G$ 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 $\{w_1, \dots, w_s\}$ be the vertex set of H and consider the image T_1, \dots, 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} . \square

Now our main theorem is established by a simple induction:

Theorem 6 *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 $M(\mathcal{C}_{k-1})$, by Theorem 5 we can compute $M(\mathcal{C}_{k-1}^{\text{apex}})$ and using Lemma 5 we can compute the connection closure of $M(\mathcal{C}_{k-1}^{\text{apex}})$ to obtain $M(\overline{\mathcal{C}_{k-1}^{\text{apex}}}) = M(\mathcal{C}_k)$. \square

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

Corollary 1 *Let \mathcal{C} be a minor-closed graph class. Then the problem ELIMINATION DISTANCE TO EXCLUDED MINORS is FPT.*

6 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 [3].

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 [21]). The class of graphs that are disjoint unions of cliques is first-order definable but certainly not nowhere dense and so the method of Sect. 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 Sect. 3 to this class?

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Adler, I., Grohe, M., Kreutzer, S.: Computing excluded minors. In: SODA '08: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM (2008)
2. Bose, R.C.: Strongly regular graphs, partial geometries and partially balanced designs. *Pac. J. Math.* **13**(2), 389–419 (1963)
3. Bulian, J., Dawar, A.: Graph isomorphism parameterized by elimination distance to bounded degree. In: Parameterized and Exact Computation—9th International Symposium, IPEC: Wroclaw, Poland, September 10–12, 2014. Revised Selected Papers **2014**, pp. 135–146 (2014)
4. Cai, L.: Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.* **58**, 171–176 (1996)
5. Dawar, A.: Finite model theory on tame classes of structures, MFCS, Lecture Notes in Computer Science, vol. **4708**, pp. 2–12. Springer (2007)
6. Dawar, A.: Homomorphism preservation on quasi-wide classes. *J. Comput. Syst. Sci.* **76**(5), 324–332 (2010)
7. Dawar, A., Grohe, M., Kreutzer, S.: Locally excluding a minor. In: Proceedings of 22nd IEEE Symposium on Logic in Computer Science, pp. 270–279 (2007)
8. Diestel, R.: Graph Theory. Springer, Berlin (2000)
9. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Berlin (2012)
10. Ebbinghaus, H.-D., Flum, J.: Finite Model Theory, 2nd edn. Springer, Berlin (1999)
11. Flum, J., Grohe, M.: Fixed-parameter tractability, definability, and model-checking. *SIAM J. Comput.* **31**(1), 113–145 (2001)
12. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Berlin (2006)
13. Fomin, F.V., Lokshtanov, D., Misra, N., Saurabh, S.: Planar F-deletion: approximation, kernelization and optimal FPT algorithms. In: 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS, pp. 470–479 (2012)
14. Frick, M., Grohe, M.: Deciding first-order properties of locally tree-decomposable structures. *J. ACM* **48**, 1184–1206 (2001)
15. Gajarský, J., Hliněný, P., Obdržálek, J., Ordyniak, S., Reidl, F., Rossmanith, P., Sanchez Villaamil, F., Sikdar, S.: Kernelization using structural parameters on sparse graph classes, Algorithms—ESA 2013—21st Annual European Symposium, pp. 529–540 (2013)
16. Golovach, P.A.: Editing to a graph of given degrees. In: Cygan, M., Heggernes, P. (eds.) Parameterized and Exact Computation, pp. 196–207. Springer, Berlin (2014)
17. Golovach, P.A.: Editing to a connected graph of given degrees. MFCS, vol. **8635**, Chapter 28, pp. 324–335 (2014)
18. Grohe, M., Kreutzer, S., Siebertz, S.: Deciding first-order properties of nowhere dense graphs. In: STOC '14: Proceedings of the 46th Annual ACM Symposium on Theory of Computing. ACM (2014)
19. Guo, J., Hüffner, F., Niedermeier, R.: A structural view on parameterizing problems: distance from triviality. In: Downey, R., Fellows, M., Dehne, F. (eds.) Parameterized and Exact Computation, pp. 162–173. Springer, Berlin (2004)
20. Hodges, W.: A Shorter Model Theory. Cambridge University Press, Cambridge (1997)
21. Hüffner, F., Komusiewicz, C., Moser, H., Niedermeier, R.: Fixed-parameter algorithms for cluster vertex deletion. *Theory Comput. Syst.* **47**, 196–217 (2010)
22. Marx, D.: Parameterized coloring problems on chordal graphs. *Theory Comput. Sci.* **351**, 407–424 (2006)
23. Mathieson, L.: The parameterized complexity of degree constrained editing problems. Ph.D. thesis, Durham University, Durham (2009)
24. Mathieson, L.: Graph Editing Problems with Extended Regularity Constraints. CoRR abs/1406.4718cs.CC (2015)
25. Mathieson, L., Szeider, S.: Editing graphs to satisfy degree constraints: a parameterized approach. *J. Comput. Syst. Sci.* **78**(1), 179–191 (2012)

26. Moser, H., Thilikos, D.M.: Parameterized complexity of finding regular induced subgraphs. *J. Discrete Algorithms* **7**(2), 181–190 (2009)
27. Nešetřil, J., Ossona de Mendez, P.: *Sparsity—graphs, structures, and algorithms*. Springer, Berlin (2012)
28. Nešetřil, J., de Mendez, P.O.: *Sparsity—Graphs, Structures, and Algorithms*. Springer, Berlin (2012)
29. Niedermeier, R.: *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, Oxford (2006)
30. Robertson, N., Seymour, P.D.: Graph minors. XX. Wagner’s conjecture. *J. Comb. Theory Ser. B* **92**, 325–357 (2004)
31. Stewart, I.A.: On the fixed-parameter tractability of parameterized model-checking problems. *Inf. Process. Lett.* **106**(1), 33–36 (2008)