Elimination Distance to Bounded Degree on Planar Graphs

Alexander Lindermayr
University of Bremen, Germany
linderal@uni-bremen.de

Sebastian Siebertz
University of Bremen, Germany
siebertz@uni-bremen.de

Alexandre Vigny
University of Bremen, Germany
vigny@uni-bremen.de

Abstract

We study the graph parameter *elimination distance to bounded degree*, which was introduced by Bulian and Dawar in their study of the parameterized complexity of the graph isomorphism problem. We prove that the problem is fixed-parameter tractable on planar graphs, that is, there exists an algorithm that given a planar graph G and integers d and k decides in time $f(k,d) \cdot n^c$ for a computable function f and constant c whether the elimination distance of G to the class of degree d graphs is at most k.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Fixed parameter tractability

Keywords and phrases Elimination distance, parameterized complexity, structural graph theory

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.65

Related Version `https://arxiv.org/abs/2007.02413`

1 Introduction

Structural graph theory offers a wealth of parameters that measure the complexity of graphs or graph classes. Among the most prominent parameters are *treedepth* and *treewidth*, which intuitively measure the resemblance of graphs with stars and trees, respectively. Other commonly studied structurally restricted graph classes are the class of *planar graphs*, classes that exclude a fixed graph as a *minor* or *topological minor*, classes of *bounded expansion* and *nowhere dense* classes.

Once we have gained a good understanding of a graph class $\mathscr{C}$, it is natural to study classes whose members are *close* to graphs in $\mathscr{C}$. One of the simplest measures of distance to a graph class $\mathscr{C}$ is the number of vertices or edges that one must delete (or add) to a graph G to obtain a graph from $\mathscr{C}$. Guo et al. [11] formalized this concept under the name *distance from triviality*. For example, the size of a minimum vertex cover is the distance to the class of edgeless graphs and the size of a minimum feedback vertex set is the distance to the class of forests. More generally, for a graph G, a vertex set X is called a *c-treewidth modulator* if the treewidth of $G - X$ is at most c, hence, the size of a c-treewidth modulator corresponds to the distance to the class of graphs of treewidth at most c. This concept was introduced and studied by Gajarský et al. in [9].

The *elimination distance* to a class $\mathscr{C}$ of graphs measures the number of recursive deletions of vertices needed for a graph G to become a member of $\mathscr{C}$. More precisely, a graph G has elimination distance 0 to $\mathscr{C}$ if $G \in \mathscr{C}$, and otherwise elimination distance $k+1$, if in every connected component of G we can delete a vertex such that the resulting graph has elimination distance k to $\mathscr{C}$. Elimination distance was introduced by Bulian and Dawar [3] in their study of the parameterized complexity of the graph isomorphism problem.

Elimination distance naturally generalizes the concept of treedepth, which corresponds to the elimination distance to the class $\mathscr{C}_0$ of edgeless graphs. The parameter also has very nice algorithmic applications. On the one hand, small elimination distance to a class $\mathscr{C}$ on which efficient algorithms for certain problems are known to exist, may allow to lift the applicability of these algorithms to a larger class of graphs. For example, Bulian and Dawar [3] showed that the graph isomorphism problem is fixed-parameter tractable when parameterized by the elimination distance to the class $\mathscr{C}_d$ of graphs with maximum degree bounded by d, for any fixed integer d. Recently, Hols et al. [12] proved the existence of polynomial kernels for the vertex cover problem parameterized by the size of a deletion set to graphs of bounded elimination distance to different classes of graphs.

On the other hand, it is an interesting algorithmic question by itself to determine the elimination distance of a given graph G to a class $\mathscr{C}$ of graphs. It is well known (see e.g. [1, 14, 15]) that computing treedepth, i.e. elimination distance to $\mathscr{C}_0$, is fixed-parameter tractable. More precisely, we can decide in time $f(k) \cdot n$ whether an n-vertex graph G has treedepth at most k. Bulian and Dawar proved in [4] that computing the elimination distance to any minor-closed class $\mathscr{C}$ is fixed-parameter tractable when parameterized by the elimination distance. They also raised the question whether computing the elimination distance to the class $\mathscr{C}_d$ of graphs with maximum degree at most d is fixed-parameter tractable when parameterized by the elimination distance and d. Note that this question is not answered by their result for minor-closed classes, since $\mathscr{C}_d$ is not closed under taking minors.

For $k, d \in \mathbb{N}$, we denote by $\mathscr{C}_{k,d}$ the class of all graphs that have elimination distance at most k to $\mathscr{C}_d$. It is easy to see that for every fixed k and d we can formulate the property that a graph is in $\mathscr{C}_{k,d}$ by a sentence in monadic second-order logic (MSO). By the famous theorem of Courcelle [6] we can test every MSO-property φ in time $f(|\varphi|, t) \cdot n$ on every n-vertex graph of treewidth t for some computable function f. Hence, we can decide for every n-vertex graph G of treewidth t whether $G \in \mathscr{C}_{k,d}$ in time $f(k, d, t) \cdot n$ for some computable function f. However, for $d \geq 3$ already the class $\mathscr{C}_d$ has unbounded treewidth, and so the same holds for $\mathscr{C}_{k,d}$ for all values of k. Thus, Courcelle's Theorem cannot be applied to derive fixed-parameter tractability of the problem in full generality.

On the other hand, it is easy to see that the graphs in $\mathscr{C}_{k,d}$ exclude the complete graph K_{k+d+2} as a topological minor, and hence, for every fixed k and d, the class $\mathscr{C}_{k,d}$ in particular has bounded expansion and is nowhere dense. We can efficiently test first-order (FO) properties on bounded expansion and nowhere dense classes [8, 10], however, first-order logic is too weak to express the elimination distance problem. This follows from the fact that first-order logic is too weak to express even connectivity of a graph or to define connected components.

While we are unable to resolve the question of Bulian and Dawar in full generality, in this work we initiate the quest of determining the parameterized complexity of elimination distance to bounded degree graphs for restricted classes of inputs. We prove that for every n-vertex graph G that excludes K_5 as a minor (in particular for every planar graph) we can test whether $G \in \mathscr{C}_{k,d}$ in time $f(k, d) \cdot n^c$ for a computable function f and constant c. Hence, the problem is fixed-parameter tractable with parameters k and d when restricted to K_5-minor-free graphs.

▶ **Theorem 1.1** (Main result). *There is an algorithm that for a K_5-minor-free input graph G with n vertices and integers k and d, decides in time $f(k, d) \cdot n^c$ whether G belongs to $\mathscr{C}_{k,d}$, where f is a computable function and c is a constant.*

Observe that the result is not implied by the result of Bulian and Dawar for minor-closed classes, as the K_5-minor-free subclass of $\mathscr{C}_d$ is not minor-closed. It is natural to consider as a next step classes that exclude some fixed graph as a minor or as a topological minor, and finally to resolve the problem in full generality.

To solve the problem on K_5-minor-free graphs we combine multiple techniques from parameterized complexity theory and structural graph theory. First, we use the fact that the property of having elimination distance at most k for fixed k is MSO definable, and hence efficiently solvable by Courcelle's Theorem on graphs of bounded treewidth. If the input graph G has small treewidth, we can hence solve the instance by Courcelle's Theorem.

If G has large treewidth, we distinguish two cases. In the first case, there exist no vertices of degree greater than $k+d$. In this case, we use the fact that G has large treewidth to conclude that it contains a large grid minor [16]. This in turn enables us to find an irrelevant vertex, that is, a vertex whose deletion does not change containment in $\mathscr{C}_{k,d}$. By iteratively removing irrelevant vertices until this is no longer possible we arrive at an instance of small treewidth. The irrelevant vertex technique was introduced in [17] and is by now a standard technique in parameterized algorithms, see [18] for a survey.

In the second case, there exist vertices of degree larger than $k+d$. Denote by R the set of all these vertices. We show that by contracting all components of $G-R$ we get a graph of bounded treedepth (and hence of bounded treewidth). We furthermore show that for each of the contracted components we can compute a *connectivity pattern* from a finite list of possible connectivity patterns that describes what happens if vertices from the component are (recursively) deleted. To compute the connectivity pattern, we again apply an irrelevant vertex argument inside the components. This part of the reasoning is technically quite involved. Once all connectivity patterns are computed, we can formulate containment in $\mathscr{C}_{k,d}$ over a colored graph of bounded treedepth in MSO, which can again be efficiently evaluated by Courcelle's Theorem. After fixing our notation in Section 2, we provide the details of the proofs in Section 3 and Section 4.

2 Preliminaries

A *graph* G consists of a set of vertices $V(G)$ and a set of edges $E(G)$. We assume that graphs are finite, simple and undirected, and we write $\{u,v\}$ for an edge between the vertices u and v. For a set of vertices $S \subseteq V(G)$, we denote the subgraph of G induced by the vertices $V(G) \setminus S$ by $G-S$. If $S=\{a\}$, we write $G-a$.

A *partial order* on a set V is a binary relation $\leq$ on V that is reflexive, anti-symmetric and transitive. A set $W \subseteq V$ is a *chain* if it is totally ordered by $\leq$. If $\leq$ is a partial order on V, and for every element $v \in V$ the set $V_{\leq v} \coloneqq \{u \in V \mid u \leq v\}$ is a chain, then $\leq$ is a *tree order*. Note that the covering relation of a tree order is not necessarily a tree, but may be a forest. An *elimination order* on a graph G is a tree order $\leq$ on $V(G)$ such that for every edge $\{u,v\} \in E(G)$ we have either $u \leq v$ or $v \leq u$. The *depth* of a vertex v in an order $\leq$ is the size of the set $V_{<v} \coloneqq \{u \in V \mid u < v\}$. The *depth* of an order $\leq$ is maximal depth among all vertices.

The *treedepth* of a graph G is defined recursively as follows.

$$\mathrm{td}(G) = \begin{cases} 0 & \text{if } G \text{ is edgeless,} \\ 1+\min\{\mathrm{td}(G-v) \mid v \in V(G)\} & \text{if } G \text{ is connected and not edgeless,} \\ \max\{\mathrm{td}(H) \mid H \text{ connected component of } G\} & \text{otherwise.} \end{cases}$$

A graph G has treedepth at most k if and only if there exists an elimination order on G of depth at most k. If the longest path in G has length k, then its treedepth is bounded by k and an elimination order of at most this depth can be found in linear time by a depth-first-search.

Elimination distance to a class $\mathscr{C}$ naturally generalizes the concept of treedepth. Let $\mathscr{C}$ be a class of graphs. The *elimination distance* of G to $\mathscr{C}$ is defined recursively as

$$\mathrm{ed}_{\mathscr{C}}(G) = \begin{cases} 0 & \text{if } G \in \mathscr{C}, \\ 1 + \min\{\mathrm{ed}_{\mathscr{C}}(G - v) \mid v \in V(G)\} & \text{if } G \notin \mathscr{C} \text{ and } G \text{ is connected}, \\ \max\{\mathrm{ed}_{\mathscr{C}}(H) \mid H \text{ connected component of } G\} & \text{otherwise.} \end{cases}$$

We denote by $\mathscr{C}_d$ the class of all graphs of maximum degree at most d and by $\mathscr{C}_{k,d}$ the class of all graphs with elimination distance at most k to $\mathscr{C}_d$. Note for instance that $\mathrm{td}(G) = k$ if and only if $G \in \mathscr{C}_{k,0}$. We write $\mathrm{ed}_d(G)$ for $\mathrm{ed}_{\mathscr{C}_d}(G)$.

▶ **Definition 2.1** (Definition 4.2 of [3]). *A tree order $\leq$ on G is an* elimination order to degree d *for G if for every $v \in V(G)$ the set $S_v \coloneqq \{u \in G \mid \{u, v\} \in E(G), u \not\leq v \text{ and } v \not\leq u\}$ satisfies either:*

- *$S_v = \emptyset$ or*
- *v is $\leq$-maximal, $|S_v| \leq d$, and for all $u \in S_v$, we have $\{w \mid w < u\} = \{w \mid w < v\}$.*

A more general notion of elimination order to a class $\mathscr{C}$ was given in the dissertation thesis of Bulian [2], which is however not needed in this generality for our purpose.

▶ **Proposition 2.2** (Proposition 4.3 in [3]). *A graph G satisfies $\mathrm{ed}_d(G) \leq k$ if, and only if, there exists an elimination order to degree d of depth k for G.*

The following lemma is easily proved by induction on k.

▶ **Lemma 2.3.** *For every graph G and elimination order $\leq$ to degree d for G, we can compute in polynomial time an elimination order $\preceq$ to degree d for G, with depth not larger than the depth of $\leq$, and with the additional property that for every $v \in V(G)$, if C, C' are distinct connected components of $G - V_{\preceq v}$ (or of G), then the vertices of C and C' are incomparable with respect to $\preceq$.*

Let G be a graph. A graph H with vertex set $\{v_1, \ldots, v_n\}$ is a *minor* of G, written $H \preceq G$, if there are connected and pairwise vertex disjoint subgraphs $H_1, \ldots, H_n \subseteq G$ such that if $\{v_i, v_j\} \in E(H)$, then there are $w_i \in V(H_i)$ and $w_j \in V(H_j)$ such that $\{w_i, w_j\} \in E(G)$. We call the subgraph H_i the *branch set* of the vertex v_i in G. If G is a graph and $\mathcal{H} = \{H_1, \ldots, H_n\}$ is a set of pairwise vertex disjoint subgraphs of G, then the graph H with vertex set $\{v_1, \ldots, v_n\}$ and edges $\{v_i, v_j\} \in E(H)$ if and only if there is an edge between a vertex of H_i and a vertex of H_j in G, the *minor induced by* $\mathcal{H}$. If $\bigcup_{1 \leq i \leq n} V(H_i) = V(G)$, then we call $\mathcal{H}$ a *minor model of H* that *subsumes all vertices* of G.

We denote by K_t the complete graph on t vertices. We denote by $G_{m,n}$ the grid with m rows and n columns, that is, the graph with vertex set $\{v_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$ and edges $\{v_{i,j}, v_{i',j'}\}$ for $|i - i'| + |j - j'| = 1$.

For our purpose we do not have to define the notion of treewidth formally. It is sufficient to note that if a graph G contains an $n \times n$ grid as a minor, then G has treewidth at least n and vice versa, that large treewidth forces a large grid minor, as stated in the next theorem.

▶ **Theorem 2.4** (Excluded Grid Theorem). *There exists a function g such that for every integer $n \geq 1$, every graph of treewidth at least $g(n)$ contains the $n \times n$ grid as a minor. Furthermore, such a grid minor can be computed in polynomial time.*

The theorem was first proved by Robertson and Seymour in [16]. Improved bounds and corresponding efficient algorithms were subsequently obtained. We refer to the work of Chuzhoy and Tan [5] for the currently best known bounds on the function g and further pointers to the literature concerning efficient algorithms.

The second black-box we use is Courcelle's Theorem, stating that we can test MSO properties efficiently on graphs of bounded treewidth. We use standard notation from logic and refer to the literature for all undefined notation, see e.g. [13].

▶ **Theorem 2.5** (Coucelle's Theorem [6]). *There exists a function f such that for every MSO-sentence φ and every n-vertex graph G of treewidth t we can test whether $G \models \varphi$ in time $f(|\varphi|, t) \cdot n$.*

3 K_5-minor-free graphs of small degree

In this section we show how to handle the case of K_5-minor free graphs of small degree. We prove the following theorem.

▶ **Theorem 3.1.** *There exists a computable function f and constant c such that for all integers k and d and every K_5-minor-free n-vertex graph G of maximum degree at most $d+k$, we can test whether $G \in \mathscr{C}_{k,d}$ in time $f(k,d) \cdot n^c$.*

▶ **Definition 3.2.** *Let G be a K_5-minor-free graph and let $k, d \in \mathbb{N}$. Assume there exists a minor model $\mathcal{G}_{m,m} = \{H_{i,j} \mid 1 \leq i,j \leq m\}$ (for $m \geq 4k+5$) that subsumes all vertices of G and that induces a supergraph of the grid $G_{m,m}$. We call the branch set $H_{i,j}$ (k,d)-safe if $2k+3 \leq i,j \leq m-2k-3$ and if $H_{i',j'}$ contains no vertex of degree at least $d+1$ (in G) for $|i-i'|, |j-j'| \leq 2k+2$.*

▶ **Lemma 3.3.** *Let G be a K_5-minor-free graph and let $k, d \in \mathbb{N}$. Assume there exists a minor model $\mathcal{G}_{m,m} = \{H_{i,j} \mid 1 \leq i,j \leq m\}$ (for $m \geq 4k+5$) that subsumes all vertices of G and that induces a supergraph of the grid $G_{m,m}$. Assume $H_{i,j}$ is (k,d)-safe. Let $a \in V(H_{i,j})$, let $B \subseteq V(G) \setminus \{a\}$ with $|B| \leq k$ and let $x, y \in V(G) \setminus (B \cup \{a\})$ be of degree at least $d+1$. Then x and y are connected in $G - B$ if and only if x and y are connected in $G - B - a$.*

Proof. Assume that x and y are connected in $G - B$ and let P be a path witnessing this. Assume that this path contains the vertex a.

We define sets X_ℓ for $0 \leq \ell \leq k+1$ of branch sets as follows. Let X_0 be the set consisting only of $H_{i,j}$ and for $\ell \geq 1$ let X_ℓ be the set of all $H_{i',j'}$ with $2\ell - 1 \leq |i-i'|, |j-j'| \leq 2\ell$ that do not already belong to $X_{\ell-1}$. The sets X_ℓ are the borders (of thickness 2) of the $(4\ell+1) \times (4\ell+1)$-subgrid around $H_{i,j}$. Observe that the vertices x and y do not belong to any of the X_ℓ, as $H_{i,j}$ is (k,d)-safe by assumption. For $0 \leq \ell \leq k+1$ let Y_ℓ be the subgraph of G induced by the vertices of X_ℓ.

We claim that for $1 \leq \ell \leq k+1$, the sets Y_ℓ are connected sets that separate a from x and analogously a from y. Clearly, the Y_ℓ are connected. Now observe that there is no edge between a vertex of $\bigcup_{0 \leq i \leq \ell-1} Y_i$ and a vertex of $G - \bigcup_{0 \leq i \leq \ell} Y_i$. The existence of such a connection would create a K_5 minor, see [7, Figure 7.10] or Figure 1. Hence, any path between a and x (or y) must pass through Y_ℓ.

As $|B| \leq k$, there is one Y_ℓ with $1 \leq \ell \leq k+1$ that does not intersect B. Let u be the first vertex that P visits on Y_ℓ on its way from x to a and let v be the last vertex that P visits on Y_ℓ on its way from a to y. As Y_ℓ is connected, we can reroute the subpath between u and v through Y_ℓ and thereby construct a path between x and y in $G - B - a$. ◀

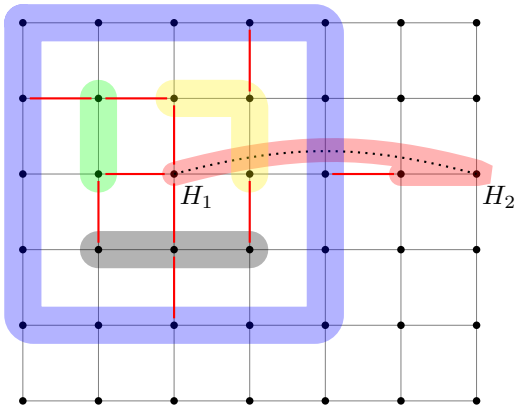

Figure 1 Construction of a K_5 minor as soon as a branch set (here H_1) connected to a branch set (here H_2) that is at distance more than 2 in the grid [7, Figure 7.10]. The blue part marks the "outer part" of the border of thickness 2, the "inner part" decomposes into a green, yellow and gray part.

Note that in the proof it is important that all vertices belong to some branch set. Otherwise, we could have a vertex x in G that does not belong to any branch set while being adjacent to $H_{i,j}$ and the whole argumentation would fail.

▶ **Corollary 3.4.** *Let G be a K_5-minor-free graph and let $k, d \in \mathbb{N}$. Assume there exists a minor model $\mathcal{G}_{m,m} = \{H_{i,j} \mid 1 \leq i, j \leq m\}$ (for $m \geq 4k+5$) that subsumes all vertices of G and that induces a supergraph of the grid $G_{m,m}$. Assume $H_{i,j}$ is (k,d)-safe. Then every vertex $a \in H_{i,j}$ is* irrelevant*, i.e., $G - a \in \mathscr{C}_{k,d}$ if and only if $G \in \mathscr{C}_{k,d}$.*

Proof. Let $H \coloneqq G - a$. We have to prove that $H \in \mathscr{C}_{k,d}$ implies $G \in \mathscr{C}_{k,d}$. Hence, assume $H \in \mathscr{C}_{k,d}$. Let $\leq_H$ be an elimination order to degree d of height k for H. We also assume that $\leq_H$ satisfies the property of Lemma 2.3, that is, for every $v \in V(G)$, if C, C' are distinct connected components of $G - V_{\leq_H v}$, then the vertices of C and C' are incomparable with respect to $\leq_H$. Let A_0 be the connected component of G containing a. Note that A_0 may break into multiple connected components in $H = G - a$.

For $1 \leq i \leq k$, we define inductively:

- m_i as the unique $\leq_H$-minimal element of $A_{i-1} \setminus \{a\}$ (if it exists) such that there exists a vertex v with $m_i \leq_H v$ and of degree at least $d+1$ in $H[A_{i-1} \setminus \{a\}]$. If there is no such element m_i, the process stops.

  Let us prove that there is at most one candidate for m_i. Assume that there are incomparable m and m' satisfying these conditions. This means that there are vertices v and v' of degree at least $d+1$ in $A_{i-1} \setminus \{a\}$ (hence of degree at least $d+1$ in G) with $m \leq_H v$ and $m' \leq_H v'$. Note that we have $m \not\leq_H v'$ because $\leq_H$ is a tree order. By Lemma 2.3, we have that v, v' are both in $A_{i-1} \setminus \{a\}$, i.e. in the connected component of a in $G - \{m_1, \ldots, m_{i-1}\}$ (it follows also by induction that $\{m_1, \ldots, m_{i-1}\} = V_{\leq_H m_{i-1}}$, hence we may apply the lemma). Hence, v and v' are connected in $G - \{m_1, \ldots, m_{i-1}\}$. With Lemma 3.3, we also have that v and v' are connected in $G - \{a, m_1, \ldots, m_{i-1}\}$. We take a witness path from v to v'. Since $m \leq_H v$ and $m \not\leq_H v'$, this path must contain two adjacent vertices w, w' with $m \leq_H w$ and $m \not\leq_H w'$. This contradicts the fact that $\leq_H$ is an elimination order satisfying the property of Lemma 2.3. Therefore, there is at most one possible such m_i.

We define

- $T_i := \{v \in A_{i-1} \ : \ m_i \not\leq_H v\}$

and

- A_i as the connected component of a in $G - \{m_1, \ldots, m_i\}$. Note that again, $A_i - \{a\}$ may be a union of connected components in $H - \{m_1, \ldots, m_i\}$.

The processes stops after at most k rounds. When the process stops, we have defined m_i, T_i and A_i up to $i = \omega$, with $\omega \leq k$ and every element in A_ω has degree at most d in $H[A_\omega]$.

We then define the new order $\leq_G$ as follows:

- for all x, y other than a and that are not in any of the T_i nor in A_ω, we have $x \leq_G y$ if and only if $x \leq_H y$,
- for all x in $A_\omega \cup \bigcup_{i \leq \omega} T_i$, we have $m_i \leq_G x$ for all $i \leq \omega$, and
- we set $m_i \leq_G a$ for all $i \leq \omega$.

Note that all the elements in A_ω, and the T_i's, together with a are $\leq_G$-maximal.

We now prove that this new order is indeed an elimination order to degree d of depth k for G. We have that $\leq_G$ is a tree order and that it has height at most k. Let us now take a vertex b and study S_b. Recall the definition of S_b from Definition 2.1. As we have two orders, we distinct S_b^G from S_b^H.

First, note that for $b = m_i$, we have $S_{m_i}^G = S_{m_i}^H = \emptyset$. So we don't have to check anything.

Then consider the case where b is a, or a neighbor of a different than m_i for all $i \leq \omega$. Then, by definition of the $(A_i)_{i \leq \omega}$, we have that $b \in A_\omega$. We also have that there is no vertex in A_ω of degree at least $d+1$ in $H[A_\omega]$. Hence $|S_b^G| \leq d$ and for any $v \in S_b^G$, we have $\{w \ : \ w <_G v\} = \{w \ : \ w <_G b\} = (m_i)_{i \leq \omega}$.

We continue with the case where b is in one of the T_i. The uniqueness of m_i implies that b has degree at most d in $H[A_i]$. Hence $|S_b^G| \leq d$ and for any $v \in S_b^G$, v also belongs to T_i, we then have that $\{w \ : \ w <_G v\} = \{w \ : \ w <_G b\} = (m_i)_{i < \omega}$.

Finally, we look at the case where b is not in A_ω, is not m_i nor in T_i for any $i \leq \omega$. In this case, we have that b cannot have neighbors in A_ω nor any of the T_i for $i \leq \omega$. To see this, assume that there is a $v \in T_i$ neighbor to b. This implies that b is in A_{i-1}, as it is connected to v, the latter being in A_{i-1}, which is the connected component of a in $G - \{m_1, \ldots, m_{i-1}\}$. As $b \notin T_i$, we have $m_i \leq_H b$ and $m_i \not\leq_H v$ which contradict that $\leq_H$ is an elimination order. This contradiction also holds if b has a neighbor in A_ω as this would imply that $b \in A_\omega$.

Therefore, in this final case, $S_b^G = S_b^H$. This also holds for any neighbor of b. Hence for any vertex v in S_b^G we have that $\{w \ : \ w <_G b\} = \{w \ : \ w <_H b\} = \{w \ : \ w <_H v\} = \{w \ : \ w <_G v\}$.

To conclude, we have that $\leq_G$ is indeed an elimination order to degree d of height k for G. This ends the proof that a is irrelevant. ◀

We can now prove the main theorem of this section.

Proof of Theorem 3.1. Let k, d be two integers and G be a connected K_5-minor free graph of maximum degree at most $k+d$. We set $h(k,d) := (4k+5)^2 \cdot (k+d)^{2(k+d)} + 4k + 5$. Let g be the function from Theorem 2.4.

Case 1. There are more than $(k+d)^{2(k+d)}$ vertices of degree at least $d+1$. We can conclude that $G \notin \mathscr{C}_{k,d}$. This is because the deletion of any vertex v can create at most $k+d$ connected components, and in each of them there are at most $k+d$ vertices whose degree can decrease by the deletion of v. Therefore by performing k elimination rounds, there will still be a vertex of degree at least $d+1$.

Case 2. The treewidth of G is bounded by $g(h(k,d))$. We use Courcelle's Theorem (Theorem 2.5) to decide whether $G \in \mathscr{C}_{k,d}$.

Case 3. We are neither in case C1) nor in case C2). Since we are not in C2), we can compute in polynomial time a grid minor of size $h(k,d)$. Furthermore we make sure that the set of branch sets subsumes all vertices. As we are not in C1), there are at most $(k+d)^{2(k+d)}$ vertices of degree at least $d+1$, and therefore, at most $(4k+5)^2 \cdot (k+d)^{2(k+d)}$ branch sets that are at distance at most $2k+3$ to a vertex of degree at least $d+1$. As $h(k,d)$ is large enough, there is a (k,d)-safe branch set which, by Corollary 3.4, implies the existence of an irrelevant vertex. We iteratively eliminate irrelevant vertices until we are in one of C1) or C2). ◀

We now do a quick complexity analysis of the algorithm. Case 1 can be solved in linear time. Case 2 requires time $f'(k, g(h(k,d))) \cdot n$, where f' is derived from the function of Theorem 2.5 and the MSO formula defining membership in $\mathscr{C}_{k,d}$. The running time in Case 3 is governed by the cost to compute a grid minor and takes time polynomial in n. Therefore, the overall complexity is $f(k,d) \cdot n^c$ for a computable function f and constant c.

4 Computing elimination distance for K_5-minor free graphs

This section is devoted to the proof of our main result: Theorem 1.1. We fix an instance (G,k), where G is K_5-minor-free.

We call a vertex $v \in V(G)$ with $d(v) \leq d$ a *blue* vertex, a vertex $v \in V(G)$ with $d < d(v) \leq k+d$ a *white* vertex and a vertex $v \in V(G)$ with $d(v) > k+d$ a *red* vertex. We denote the set of red vertices by R, the set of white vertices by W and the set of blue vertices by B. All red vertices have to be deleted in the elimination process, if this is not possible, then (G,k) is a negative instance. For all white vertices we have a choice of whether we want to delete the vertex itself or some of its neighbors. Blue vertices already satisfy the degree condition and will be only deleted if their deletion creates useful components.

▶ **Lemma 4.1.** *If $G \in \mathscr{C}_{k,d}$, then every path in G contains at most $2^k - 1$ red vertices.*

Proof. By deleting a red vertex we can only split the path into half. Hence, the number of red vertices on a path is bounded by the function that is recursively defined by $f(1) = 1$ and $f(k+1) = 2f(k) + 1$. This defines the function $f(k) = 2^k - 1$. ◀

▶ **Lemma 4.2.** *Let C be a component of $G - R$. If $G \in \mathscr{C}_{k,d}$, then there are at most $(k+d)^k$ red vertices that are neighbors of a vertex of C in G.*

Proof. By induction on k, using the fact that the deletion of a white or blue vertex can create at most $k+d$ components in G and all red vertices have to be deleted. ◀

Having these two lemmas, we can now develop our algorithm. We first merge every component C of $G - R$ into a single vertex v_C to obtain a new graph G'. If one of the new vertices v_C has degree larger than $(k+d)^k$, then we may reject the instance due to Lemma 4.2. We then compute a depth-first-search of G'. Observe that no two vertices v_C and $v_{C'}$ for distinct components C, C' of $G - R$ are adjacent in G', as otherwise C and C' would not be separate components. Hence, the search outputs and elimination order of G' of depth at most $2^{k+1} - 1$, or we may reject the instance due to Lemma 4.1. We may hence assume in the following that G' has treedepth at most $2^{k+1} - 1$.

We can therefore test whether G' belongs to $\mathscr{C}_{k,d}$ via the evaluation of an MSO sentence. However, since this graph has been obtained by merging components of $G - R$, the deletion of a node v_C in G' might require the deletion of the entire component C of $G - R$, which can have an arbitrary size. Fortunately, we do not have to delete the entire component but only to "separate" its red neighbors and to "fix" vertices of degree between $d+1$ and $d+k$ inside the component.

To do that, every non-red vertex v_C of G' will be associated with a type that describes the internal "connectivity pattern" of the $G - R$ component C it corresponds to. This is possible thanks to Lemma 4.2, which will help to bound the number of possible *connectivity patterns* with the rest of the graph. Testing whether a $G - R$ component satisfies some connectivity pattern will be performed via an algorithm that resembles the one that proves Theorem 3.1.

Let us now make this informal argumentation more concrete.

▶ **Definition 4.3.** *Let k and d be two integers, and let $p \coloneqq (k+d)^k$. By a* partition*, we mean a partition of the set $\{1, \dots, p\}$.*

A partition $P = (A_i)_{i \leq \ell}$ refines *another partition $P' = (A'_i)_{i \leq \ell'}$ if and only if for every $i \leq \ell$ there is a $j \leq \ell'$ such that $A_i \subseteq A'_j$.*

For a pair of integers (j, j'), we write $(j, j') \in P$ if there is a set A_i that contains both j and j'. We refer to such a pair of integers as being grouped *in P, as opposed to being* split *or* separated *in P. An integer j that is not grouped with any other integer is* isolated *in P.*

▶ **Definition 4.4.** *Let k and d be two integers, and let $p \coloneqq (k+d)^k$. A (k,d)-*sequence *is a sequence $(P_i, L_i, D_i)_{i \leq \ell}$ for $\ell \leq k$, where P_i and L_i are partitions of $\{1, \dots, p\}$ and D_i is a subset of $\{1, \dots, p\}$.*

Note in particular that there are a bounded number of (k,d)-sequences. Let us write H for some component of $G - R$, which according to Lemma 4.2 has at most $p \coloneqq (k+d)^k$ red neighbors. We use p unary predicates $C_1, \dots, C_p$ to mark the vertices of H that are neighbors of a red vertex (the red vertices are not vertices of H). A (k,d)-sequence for H will describe the elimination process from the point of view of H. The partition P_i will represent how the different unary predicates are linked inside of H at depth i of the elimination process. The partition L_i will represent how the unary predicates are connected outside of H and the subset D_i is the list of predicates that correspond to red vertices that have been deleted.

▶ **Definition 4.5.** *Given integers k, d, $p \coloneqq (k+d)^k$, and a graph H with maximum degree $k+d$ with p unary predicates $C_1, \dots C_p$, we say that H* satisfies *a (k,d)-sequence $S = (P_i, L_i, D_i)_{i \leq \ell}$, for some $\ell \leq k$ if there is an elimination order $\leq_o$ to degree d of depth ℓ for H that satisfies:*

- *For every $1 < i \leq \ell$ we have that P_i (resp. L_i) refines P_{i-1} (resp. L_{i-1}).*
- *For every $1 < i \leq \ell$ we have $D_{i-1} \subseteq D_i$.*
- *For every $i \leq \ell$ and every $j \in D_i$, j is isolated in L_i.*
- *If two vertices b, b' satisfy $b, b' \in C_j$ for some $j \leq p$ and b, b' are incomparable in $\leq_o$, then either b and b' are $\leq_o$-maximal and $\{w \mid w \leq_o b\} = \{w \mid w \leq_o b'\}$, or there is no vertex v and integer i such that $v \leq_o b$, $v \not\leq_o b'$, and v is at depth at most $i-1$, where $j \notin D_i$.*
- *If two vertices b, b' satisfy $b \in C_{j_1}$, $b' \in C_{j_2}$ for some $j_1, j_2 \leq p$ and b, b' are incomparable in $\leq_o$, then either b and b' are $\leq_o$-maximal and $\{w \ : \ w \leq_o b\} = \{w \ : \ w \leq_o b'\}$, or there is no vertex v and integer i such that $v \leq_o b$, $v \not\leq_o b'$, v is at depth at most $i-1$ and $(j_1, j_2) \in L_i$.*
- *For every $i \leq \ell$, two integers $(j_1, j_2) \in P_i$ if and only if there is a path in H from a vertex satisfying C_{j_1} to a vertex satisfying C_{j_2} using only maximal vertices, and vertices at* depth *at least i.*

▶ **Example 4.6.** For example, fix $d = 2$, $k > 4$ and a component with only 3 unary predicates as depicted in Figure 2.

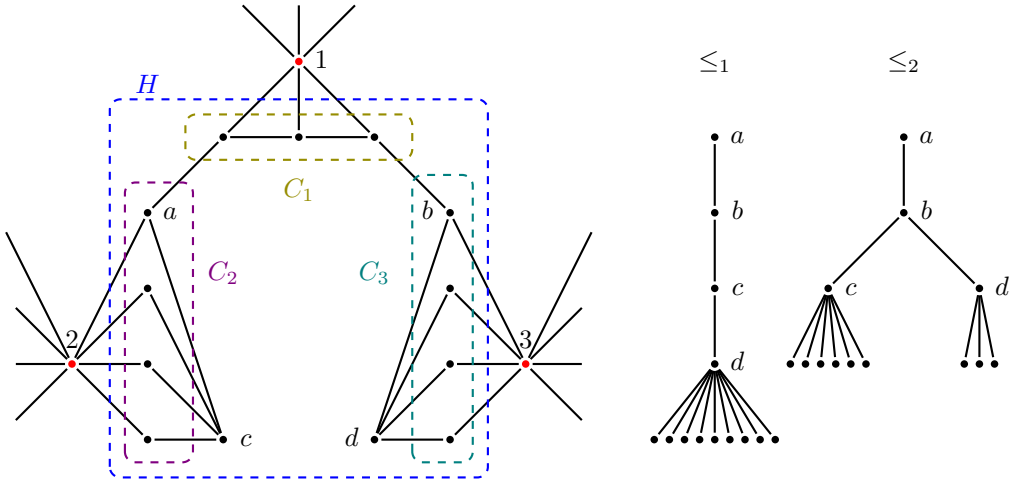

Figure 2 Example of graph and elimination orders to degree 2.

Consider now the sequences:

$S_1 := \Big(\{1,2,3\};\{1,2,3\};\emptyset\Big);\Big(\big(\{1\},\{2\},\{3\}\big);\{1,2,3\};\emptyset\Big);$

$\Big(\big(\{1\},\{2\},\{3\}\big);\{1,2,3\};\emptyset\Big);\Big(\big(\{1\},\{2\},\{3\}\big);\{1,2,3\};\emptyset\Big);$

$S_2 := \Big(\{1,2,3\};\{1,2,3\};\emptyset\Big);\Big(\big(\{1\},\{2\},\{3\}\big);\{1,2,3\};\emptyset\Big);\Big(\big(\{1\},\{2\},\{3\}\big);\{1,2,3\};\emptyset\Big);$

$S_3 := \Big(\{1,2,3\};\{1,2,3\};\emptyset\Big);\Big(\big(\{1\},\{2\},\{3\}\big);\{1,2,3\};\emptyset\Big);\Big(\big(\{1\},\{2\},\{3\}\big);\big(\{1,2\},\{3\}\big);\{3\}\Big)$

Consider the graph in Figure 2. Sequence S_1 can be satisfied if at depth 2, we are left with no internal path between the different predicates ($1, 2, 3$ are split in P_2). This is achievable by removing a and then b. Also, the order has to be an elimination order to degree 2 of depth at most 4. It is achieved by also removing c and d. This corresponds to the order $\leq_1$, which witnesses that H satisfies sequence S_1.

Sequence S_2 also asks for the deletion of all paths between the different predicates. However, we also need the order to be an elimination order to degree 2 of depth 3. So in one round something must be done to c and d. This is impossible because $(2, 3)$ are still grouped in L_3. For example, the order $\leq_2$ is not a witness because there are vertices $v \in C_2$, $u \in C_3$ with $c \leq_2 v$, $c \not\leq_2 u$, while $(2, 3)$ are still grouped in L_3, and c is at depth 2, strictly less than 3.

The third sequence is satisfied by H, as witness by $\leq_2$. This time, $(2, 3)$ are split in L_3.

▶ **Lemma 4.7.** *For all integers k, d and (k, d)-sequence S, there is an* MSO *formula Φ_S that expresses the fact that a graph H satisfies S.*

We simply have to redefine the notion of connectivity according to the (k, d)-sequence S. Two vertices are considered adjacent at depth i if they share an edge, or if they are both in some unary predicate C_j for some j that is not in D_i, or if they are respectively in some C_j, $C_{j'}$ with j, j' grouped in L_i. With this modified notion of adjacency, which is expressible by MSO, also connectivity can easily be expressed by an MSO formula. The precise MSO formula can be found in the full version of the paper.

▶ **Lemma 4.8.** *There is an algorithm that, given two integers k, d, a (k, d)-sequence S and a K_5-minor-free n-vertex graph H of degree at most $k + d$ with $p \coloneqq (k + d)^k$ many unary predicates, tests whether H satisfies S in time $f(k, d) \cdot n^c$ for some computable function f and constant c.*

The proof of this lemma technical, however, the mains ideas and techniques are already present in the proof of Theorem 3.1. The proof of the lemma can be found in the full version.

▶ **Lemma 4.9.** *Let G be a graph and let G' be the graph obtained by merging all components of $G - R$ into single vertices, which are labeled by the set of (k, d)-sequences they satisfy. Then there exists an MSO-formula that is satisfied in G' if and only if $G \in \mathscr{C}_{k,d}$.*

Note that we assume that G' as already been computed, including all of the labels. Note also that at this point we do not require G to be K_5-minor-free. To prove Lemma 4.9 we again define an appropriate notion of connectivity, using for every non red vertex the different P_i in the (k, d)-sequences that this component satisfies. We then end up with a formula that resembles the one witnessing that $\mathscr{C}_{k,d}$ is MSO definable. Details and formulas are presented in the full version of the paper.

We are now ready to prove the main result.

Proof of Theorem 1.1. Given integers k, d, and a K_5-minor free graph G, we replace it with a graph G', with unary predicates. More precisely, we define one unary predicate τ_S per (k, d)-sequence S. Every component of $G - R$ is replaced in G' by a unique node that satisfies the predicate τ_S for every sequence S satisfied by this component.

By Lemma 4.8, we can test with an FPT algorithm whether a component satisfies a given sequence S. We can therefore compute G' with an FPT algorithm. By Lemma 4.1, G' has tree-depth at most $2^{k+1} - 1$. By Lemma 4.9, testing whether G is in $\mathscr{C}_{k,d}$ can be tested with an MSO formula on G', which is therefore computed by an FPT algorithm. ◀

5 Conclusion

Our proofs show that all difficulties for elimination distance to bounded degree arise already in bounded degree graphs. We were not able to solve these difficulties in general, but rely on the additional assumption that the input graphs exclude K_5 as a minor. Hence the following problem remains open.

▶ **Open Problem 5.1.** *Is there an algorithm that, on input integers k, d, Δ and an n-vertex graph graph G of maximum degree at most Δ, tests whether G belongs to $\mathscr{C}_{k,d}$ in time $f(k, d, \Delta) \cdot n^c$ for some computable function and constant c?*

We conjecture that if such algorithm exists, it will also be possible to test whether a graph G satisfies some (k, d)-sequence. Hence we would be able to remove assumption of excluding K_5 as a minor from Lemma 4.8. If this can be done, then our proof of Theorem 1.1 follows. This yields the following conjecture:

▶ **Conjecture 5.2.** *The membership problem of $\mathscr{C}_{k,d}$ is* FPT *if and only if it is* FPT *when restricted to classes of graphs with bounded degree.*

References

1 Hans L Bodlaender, Jitender S Deogun, Klaus Jansen, Ton Kloks, Dieter Kratsch, Haiko Müller, and Zsolt Tuza. Rankings of graphs. *SIAM Journal on Discrete Mathematics*, 11(1):168–181, 1998.

2 Jannis Bulian. Parameterized complexity of distances to sparse graph classes. Technical report, University of Cambridge, Computer Laboratory, 2017.

3 Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica*, 75(2):363–382, 2016.

4 Jannis Bulian and Anuj Dawar. Fixed-parameter tractable distances to sparse graph classes. *Algorithmica*, 79(1):139–158, 2017.

5 Julia Chuzhoy and Zihan Tan. Towards tight (er) bounds for the excluded grid theorem. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1445–1464. SIAM, 2019.

6 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and computation*, 85(1):12–75, 1990.

7 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

8 Zdeněk Dvořák, Daniel Král, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *Journal of the ACM (JACM)*, 60(5):36, 2013.

9 Jakub Gajarský, Petr Hliněný, Jan Obdržálek, Sebastian Ordyniak, Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. Kernelization using structural parameters on sparse graph classes. *Journal of Computer and System Sciences*, 84:219–242, 2017.

10 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM (JACM)*, 64(3):17, 2017.

11 Jiong Guo, Falk Hüffner, and Rolf Niedermeier. A structural view on parameterizing problems: Distance from triviality. In *International Workshop on Parameterized and Exact Computation*, pages 162–173. Springer, 2004.

12 Eva-Maria C. Hols, Stefan Kratsch, and Astrid Pieterse. Elimination distances, blocking sets, and kernels for vertex cover. In *STACS*, 2020.

13 Leonid Libkin. *Elements of finite model theory*. Springer Science & Business Media, 2013.

14 Jaroslav Nešetřil and Patrice Ossona De Mendez. *Sparsity: graphs, structures, and algorithms*, volume 28. Springer Science & Business Media, 2012.

15 Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. A faster parameterized algorithm for treedepth. In *International Colloquium on Automata, Languages, and Programming*, pages 931–942. Springer, 2014.

16 Neil Robertson and Paul D Seymour. Graph minors. V. excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986.

17 Neil Robertson and PD Seymour. Graph minors. XIII. the disjoint paths problem. *J. Combin. Theory Ser. B*, 63:65–110, 1995.

18 Dimitrios M Thilikos. Graph minors and parameterized algorithm design. In *The Multivariate Algorithmic Revolution and Beyond*, pages 228–256. Springer, 2012.