

Pushing the Frontiers of Subexponential FPT Time for Feedback Vertex Set

Gaétan Berthe 

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Marin Bougeret 

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Daniel Gonçalves 

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Jean-Florent Raymond 

CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP, UMR5668, Lyon, France

Abstract

The paper deals with the FEEDBACK VERTEX SET problem parameterized by the solution size. Given a graph G and a parameter k , one has to decide if there is a set S of at most k vertices such that $G - S$ is acyclic. Assuming the Exponential Time Hypothesis, it is known that FVS cannot be solved in time $2^{o(k)}n^{\mathcal{O}(1)}$ in general graphs. To overcome this, many recent results considered FVS restricted to particular intersection graph classes and provided such $2^{o(k)}n^{\mathcal{O}(1)}$ algorithms.

In this paper we provide generic conditions on a graph class for the existence of an algorithm solving FVS in subexponential FPT time, i.e. time $2^{k^\varepsilon} \text{poly}(n)$, for some $\varepsilon < 1$, where n denotes the number of vertices of the instance and k the parameter. On the one hand this result unifies algorithms that have been proposed over the years for several graph classes such as planar graphs, map graphs, unit-disk graphs, pseudo-disk graphs, and string graphs of bounded edge-degree. On the other hand it extends the tractability horizon of FVS to new classes that are not amenable to previously used techniques, in particular intersection graphs of “thin” objects like segment graphs or more generally s -string graphs.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Theory of computation → Fixed parameter tractability; Theory of computation → Computational geometry

Keywords and phrases Subexponential FPT algorithms, geometric intersection graphs

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.26

Category Track A: Algorithms, Complexity and Games

Related Version Full Version: <https://arxiv.org/abs/2504.17708> [7]

1 Introduction

1.1 Context

Given an n -vertex graph G and a parameter $k \in \mathbb{N}$, the FEEDBACK VERTEX SET problem (FVS for short) asks whether there exists a set S of at most k vertices such that $G - S$ has no cycle. This is a fundamental decision problem in graph theory and one of Karp’s 21 NP-complete problems. Because of its hardness in a classical setting, the problem has been widely studied within the realm of parameterized complexity. This line of research aims to investigate the existence of *FPT algorithms*, i.e., algorithms that run in time $f(k) \cdot n^{\mathcal{O}(1)}$, for some computable function f . Such algorithms provide a fine-grained understanding on the time complexity of a problem and describe regions of the input space where the problem can be solved in polynomial time. Note that it is crucial here to obtain good bounds on the function f since the (potentially) super-polynomial part of the running time is confined in



© Gaétan Berthe, Marin Bougeret, Daniel Gonçalves, and Jean-Florent Raymond;
licensed under Creative Commons License CC-BY 4.0

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).

Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis

Article No. 26; pp. 26:1–26:15



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



the $f(k)$ term. In this direction it was proved that under the Exponential Time Hypothesis of Impagliazzo, Paturi and Zane, FVS does not admit an algorithm with running time $2^{o(n)}n^{O(1)}$ (see [11]). Nevertheless certain classes of graphs (typically planar graphs) have been shown to admit algorithms with running times of the form $2^{\mathcal{O}(k^\varepsilon)}n^{O(1)}$ (for some $\varepsilon < 1$), i.e., where the contribution of the parameter k is subexponential. Such algorithms are called *subexponential parameterized algorithms* and they are the topic of this paper. Given the numerous existing results on this theme, there are two main directions of research: to improve the running times in the classes where an algorithm is already known, or to extend the tractability horizon of FVS by providing more general settings where subexponential FPT algorithms exist. We are here interested by the second direction, that can be summarized by the following question.

► **Question 1.** What are the most general graph classes where FVS admits a subexponential parameterized algorithm?

Historically, a primary source of graph classes studied to make progress on the above question was geometric intersection graphs. In an *intersection graph*, each vertex corresponds to a subset of some ambient space, and two vertices are adjacent if and only if the subsets intersect. Taking the Euclidean plane as the ambient space, many graph classes can be defined by setting restrictions on the subsets used to represent the vertices. One can for instance consider intersection graphs of disks in the plane, or segments, or Jordan arcs¹. With such subsets, one defines the class of *disk graphs*, *segment graphs*, and *string graphs* respectively. It is also often the case that there are conditions dealing with all the n subsets representing the vertices of a given graph. For example, if we consider disks (resp. segments) one can ask those to have the same diameter (resp. to use at most d different slopes), and this defines the class of *unit disk graphs* (resp. *d-Dir graphs*). When considering strings, one possible property is that any string has at most d points shared with the other considered strings. This defines string graphs with *edge degree at most d*. A weaker condition is to ask every pair of the considered strings to intersect on at most s points, this defines *s-string graphs*. This class generalizes several natural classes such as planar graphs, map graphs, unit-disk graphs, segment graphs, string graphs of bounded edge-degree, and intersection graphs of α -convex bodies that exclude a fixed subgraph (see [25, 18, 3]).

For all these graph classes, FVS is NP-complete, and actually under ETH none of them admits a $2^{o(\sqrt{n})}$ -time algorithm. Indeed, this lower bound was given in [12] for induced grid graphs, which form a subclass of unit disk graphs and 2-Dir graphs. On the other hand for each of the aforementioned classes there is an algorithm solving FVS in subexponential time. More precisely, this algorithm applies to any string graph and runs in time $2^{\tilde{O}(n^{2/3})}$ [9].² Regarding subexponential parameterized algorithms, the case of unit disk graphs was settled with an algorithm whose running time matches the ETH lower bound [2]. This result uses the fact that these graphs admit vertex partitions into cliques such that each of these cliques is adjacent to only a constant number of the other cliques. Such a property does not hold for the other graph classes mentioned above. However, other techniques have been developed to deal with the other aforementioned classes such as the classes of bounded edge degree string graphs [3], contact-segment and square graphs [5], disk graphs [21, 1], or the pseudo-disk graphs [4]. Note that when dealing with classes of intersection graphs, the representation of the input (if known) could be used by the algorithm. Some of these algorithms are *robust*, meaning that the input graph G is provided using one of the classical graph data structures, where there is no indication of the intersection model of G . Because the recognition problem is difficult for most of the classes discussed above, robustness is a substantial advantage.

¹ In the following, those are called strings.

² The notation $\tilde{O}$ ignores polylogarithmic factors, i.e. we write $g(x) = \tilde{O}(f(x))$ if for some c we have $g(x) = \mathcal{O}(f(x) \cdot \log^c x)$.

1.2 Our contribution

Toward answering Question 1, in the same spirit as in [23], which introduces a generic setting where certain hitting problems can be solved in subexponential parameterized time, we identify sufficient conditions for a graph class (then said to be *nice*) to admit a subexponential parameterized algorithm for FVS. As we will see later these conditions are satisfied by several natural graph classes, some of which were not known to admit a subexponential parameterized algorithm prior to this work.

Let us now provide some intuition behind the conditions we require for a *nice* graph class. We discuss here the similarities between these conditions and classical studied properties, while the reasons why these conditions help to get a subexponential parameterized algorithm for FVS are examined in Section 2. A starting point is to review known results about the class of string graphs, which constitutes a good candidate to answer the previous question. In particular, the following results are known for string graphs. For a graph H , let us say that a graph is H -free if it does not contain H as subgraph.

► **Theorem 2** ([20],[14]). $K_{r,r}$ -free string graphs on n vertices have treewidth $\mathcal{O}(\sqrt{nr \log r})$.

► **Theorem 3** ([20]). There exists a constant c such that for $r > 0$ it holds that every $K_{r,r}$ -free string graph on n vertices has at most $cr(\log r)n$ edges.

These results³ are interesting in our case, as a simple folklore branching allows us to reduce the problem to the case where the instance (G, k) of FVS is $K_{r,r}$ -free for $r = \lceil k^\epsilon \rceil$. Thus, among the conditions required for a graph class to be nice, two of them correspond to a relaxed version of the above theorems.

Our last main condition is related to neighborhood complexity. A graph class $\mathcal{G}$ has *linear neighborhood complexity (with ratio c)* if for any graph $G \in \mathcal{G}$ and any $X \subseteq V(G)$, $|\{N(v) \cap X, v \in V(G)\}| \leq c|X|$. It is known that bounded-expansion graph classes have linear neighborhood complexity [26] as well as bounded twin-width graphs [8]. In previous work on parameterized subexponential algorithms [22, 1, 4, 5], it appeared useful that the considered graphs have the property that, if G is $K_{r,r}$ -free (or even K_r -free), then for any $X \subseteq V(G)$, $|\{N(v) \cap X, v \in V(G)\}| \leq r^{\mathcal{O}(1)}|X|$. Notice that this is slightly stronger than requiring that a class that is $K_{r,r}$ -free (or K_r -free) for a fixed r has linear neighborhood complexity, as it is important for our purpose that the dependency in r is polynomial. We point out that $K_{r,r}$ -free string graphs have bounded-expansion, hence linear neighborhood complexity, however this does not imply that the dependency in r is polynomial. Thus, our last main condition (called *bounded tree neighborhood complexity*) can be seen as a slightly stronger version of this “polynomially dependent” neighborhood complexity. Let us now proceed to the formal definitions.

► **Definition 4.** We say that a graph class $\mathcal{G}$ has bounded tree neighborhood complexity (with parameters α, f_1, f_2) if there exist an integer α and two polynomial functions f_1, f_2 such that the following conditions hold. For every r , every $K_{r,r}$ -free graph $G \in \mathcal{G}$, every set $A \subseteq V(G)$ and every family $\mathcal{T}$ of disjoint non-adjacent⁴ vertex subsets of $G - A$, each inducing a tree:

³ Note that an error was found in the proof of the above results in [20], but a claim by the author was made that the proof can be corrected in the case of string graphs (see [10]). Moreover the earlier bound in [24] yields similar results, up to logarithmic factors.

⁴ Two vertex subsets are *non-adjacent* in a graph if there is no edge from one to the other, see Subsection 1.3.

- $|\{N_A(T), T \in \mathcal{T}\}| \leq f_1(r)|A|^\alpha$, where $N_A(T)$ denotes the neighbors of the vertices of T in A , and
- $|\{N_A(T), T \in \mathcal{T}\}| \leq f_2(r, p, m)|A|$, where p and m denote the maximum over all $T \in \mathcal{T}$ of $|N_A(T)|$ and $|T|$ respectively.

► **Definition 5.** An hereditary graph class $\mathcal{G}$ is nice (for parameters $\alpha, f_1, f_2, \delta, f, d$) if all the following conditions hold:

- C1. $\mathcal{G}$ is stable by contraction of an edge between degree-two vertices that do not belong to a triangle.
- C2. $\mathcal{G}$ has bounded tree neighborhood complexity (for some parameters α, f_1, f_2).
- C3. There exist $\delta < 1$ and a constant f_r that depends polynomially in r such that for any $K_{r,r}$ -free graph $G \in \mathcal{G}$, $\text{tw}(G) = \mathcal{O}(f_r \cdot n^\delta)$.
- C4. There is a constant d_r that depends polynomially in r such that for any $K_{r,r}$ -free graph $G \in \mathcal{G}$, $|E(G)| \leq d_r \cdot |V(G)|$. Without loss of generality we will assume $d_r \geq r$.

Our main result is the following.

► **Main Theorem.** For every nice hereditary graph class $\mathcal{G}$ there is a constant $\eta < 1$ such that FVS can be solved in $\mathcal{G}$ in time $2^{\mathcal{O}(k^\eta)} \cdot n^{\mathcal{O}(1)}$.

Actually we provide a single generic algorithm for all nice classes (which requires the values of α, f_1, f_2, d , but not f and δ). The techniques used to prove the above result are discussed in Section 2. For the time being, let us focus on consequences. As hinted above, being nice is a natural property shared by several well-studied classes of graphs. In particular we show that it is the case for s -string graphs and pseudo-disk graphs, hence we have the following applications.

► **Corollary 6.** There exists $\eta < 1$, such that for all s there is a robust parameterized subexponential algorithm solving FVS in time $2^{\tilde{\mathcal{O}}(s^{\mathcal{O}(1)} k^\eta)} n^{\mathcal{O}(1)}$ for n -vertex s -string graphs.

► **Corollary 7.** There exists $\eta < 1$, such that there is a robust parameterized subexponential algorithm solving FVS in time $2^{\mathcal{O}(k^\eta)} n^{\mathcal{O}(1)}$ for n -vertex pseudo-disk graphs.

Observe that the two corollaries above encompass a wide range of classes of geometric intersection graphs for which subexponential parameterized algorithms have been given in previous works such as planar graphs, map graphs, unit-disk graphs, disk graphs, or more generally pseudo-disk graphs, and string graphs of bounded edge-degree. In this sense our main result unifies the previous algorithms.

Also, it captures new natural classes such as segment graphs, or more generally s -string graphs, where previous tools were not sufficient (as discussed in Section 2). We point out that before this work, the existence of subexponential parameterized algorithm for FVS was open even for the very restricted class of 2-DIR graphs.

Generality has a cost and the running time bound we obtain for pseudo-disk graphs is worst than the one obtained in [4], which heavily relied on the input pseudo-disk representation. On the other hand our algorithm has the extra property of being robust (i.e., it does not require a geometric representation) which is a relevant advantage for those classes of intersection graphs where computing a representation is difficult.

1.3 Basic notations and organisation of the paper

In this paper logarithms are binary and all graphs are non-oriented and simple. Unless otherwise specified we use standard graph theory terminology, as in [13] for instance. For a graph G , and $v \in V(G)$, we denote $N_G(v)$ the neighbors of v . We omit the subscript when it is clear from the context. For $A \subseteq V(G)$, we use the notation $N(A) = (\cup_{v \in A} N(v)) \setminus A$ and denote $G[A]$ the subgraph induced by G on A . For $v \in V(G)$ and $B \subseteq V(G)$, we denote $N_B(v) = N(v) \cap B$ and for $A \subseteq V(G)$ we denote $N_B(A) = N(A) \cap B$, with the additional notation $d_B(A) = |N_B(A)|$. For a graph H we say that G is H -free if H is not a subgraph of G . Two disjoint vertex subsets or subgraphs Z, Z' of a graph G are said to be *non-adjacent* (in G) if there is no edge in G with one endpoint in Z and the other in Z' .

Organisation

In Section 2 we explain why the approaches developed for other intersection graph classes (mainly in the papers [1, 3, 4, 21]) do not apply here and present the main ideas behind our algorithm. The presentation uses a case study where we are given a small set M such that $G - M$ is a forest of paths. This simplifies the arguments but is complex enough to justify the restrictions we put on the classes we consider. In Section 3 we provide applications of our main theorem by showing in particular that s -string graphs are nice. Finally, in Section 4 we discuss open problems and possible extensions of the approach developed here. Due to space constraints, the description of the algorithm in the general case and its analysis can be found in the full version of the paper.

2 Our techniques

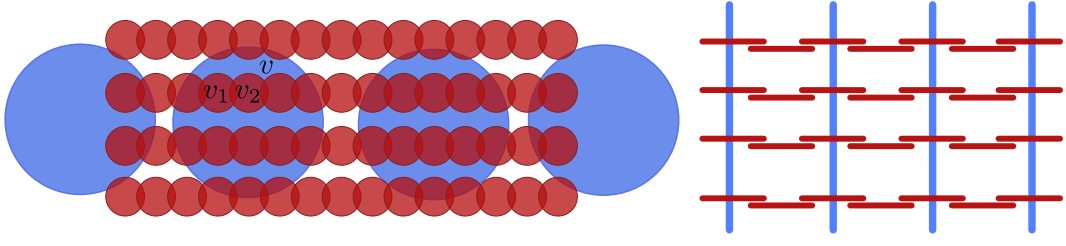
2.1 Why bidimensionality fails and differences with classes of “fat” objects

Even if our goal is to abstract from a specific graph class, let us consider in this section the class of 2-DIR graphs, corresponding to the intersection graphs of vertical or horizontal segments in the plane. As these objects are non “fat”⁵ and can cross (unlike pseudo-disks), this class constitutes a good candidate to exemplify the difficulties.

A common approach is as follows. Given an instance (G, k) of FVS, we compute first in polynomial time a 2-approximation, implying that we either detect a no-instance, or define a set M with $|M| \leq 2k$ and such that $G - M$ is a forest. The goal is then to reduce, using kernelization or subexponential branching rules, to equivalent instances (G', k') with small treewidth, that is $\text{tw}(G') = \mathcal{O}(k^{1-\varepsilon})$. As FVS can be solved in $2^{\mathcal{O}(\text{tw}(G'))} n^{\mathcal{O}(1)}$ using a classical dynamic programming approach, we get a subexponential parameterized algorithm. Thus, one has to find a way to destroy in G the obstructions preventing a small treewidth. A first type of obstructions is K_r and $K_{r,r}$, which are easy to handle as there are folklore subexponential branchings when $r = k^\varepsilon$. Now, one can see the hard part as destroying $K_{r,r}$ hidden (as a minor for example) (see Figure 1).

A point that seems crucial to us is the following. In intersection graphs of “fat objects” (like disks, squares, or pseudo-disks more generally), the “locally non planar structure” when an object (vertex v in Figure 1) is “traversed” (by v_1, v_2 in Figure 1) comes to the price

⁵ A connected regions R of the plane is said to be α -fat if the radius of smallest disk enclosing R is at most α times larger than the radius of the largest disk enclosed in S . A family of regions of the plane is then said to be *fat* if there exists α such that all the elements of the family are α -fat.



■ **Figure 1** Example of a $K_{r,r}$ contained as a minor for $r = 4$ in a disk graph (left) and a 2-DIR graph (right). In the case of disk graph, v has a matching of size $r - 2$ in its neighborhood, forming a triangle bundle, which can be exploited to branch. The set M are depicted in blue. For the 2-DIR graph, the vertices of the long paths are represented by segments with small variation in their height and not intersecting for better clarity, but are in fact on the same level and intersecting.

of an edge ($\{v_1, v_2\}$) in the neighborhood of v . Thus, the presence of a large $K_{r,r}$ as a minor implies that a large matching E_v (of size $\Omega(r)$) will appear in the neighborhood of a vertex v . However, as $G[\{v\} \cup E_v]$ (called a triangle bundle in [22]) contains r triangles pairwise intersecting on exactly one vertex v , the set $\{v\} \cup E_v$ is a good structure to perform a subexponential branching for FVS. Indeed, [21] proposed a “virtual branching” to handle this structure by either taking v in the solution, or absorbing E_v in M , implying then that the parameter virtually decreases by $|E_v|$ as a solution which does not contain v has to hit all these edges, even if we cannot branch to determine which are exactly the vertices in the solution.

Once no more virtual branching is possible on large triangle bundles, they obtain by some additional specialized techniques that any vertex in M is such that $N_{V(G) \setminus M}(v)$ is an independent set. Then, it is proved ([21], Corollary 1.1) that in a disk graph where for any $v \in M$, $N_{V(G) \setminus M}(v)$ is an independent set, and where there does not exist a vertex in $V(G) \setminus M$ whose neighborhood is contained in M , then $tw(G) = \mathcal{O}\left(\sqrt{|M|}\omega(G)^{2.5}\right)$, where $\omega(G)$ denotes the maximum size of a clique in G . This no longer holds for 2-DIR graphs: the family of pairs (G, M) depicted on the right of Figure 1 is indeed a counter example as they respect the conditions, have $\omega(G) = 2$, but $tw(G) = \Omega(|M|)$.

More generally, the role of the size of a matching in the neighborhood was studied in [5] which shows how subexponential parameterized algorithms can be obtained for graph classes having the “almost square grid minor property” (ASQGM), corresponding informally⁶ to $tw(G) = \mathcal{O}(\omega(G)^{\mathcal{O}(1)}\mu_N(G) \boxplus(G))$ where $\mu_N(G)$ is the maximum size of a matching in a neighborhood of a vertex, and $\boxplus(G)$ is the largest size of a grid contained as a minor in G . The previous counter example shows that 2-DIR does not have the ASQGM property, implying that we need another approach to handle them.

2.2 A simpler case study: when trees are only paths

To simplify the arguments, but still understand why properties in Definition 5 of a nice graph class are needed, let us assume that the forest $G - M$ only contains paths $(P_i)_i$, and use the example of segment graphs. This case remains challenging as a large $K_{r,r}$ can still be hidden as a minor (as in Figure 1), and we need to destroy it in order to reduce the treewidth of the graph. To keep notations simple, we use the notation $poly(\cdot)$ to denote a polynomial

⁶ In the correct definition $\mu_N(G)$ is replaced by a slightly more technical parameter.

dependency on the parameter, and thus we do not try to compute tight formulas depending on the polynomial f_1, f_2, f, d given in the definition of a nice class. We assume that we performed folklore branching and that we are left with a $K_{r,r}$ -free graph G for $r = k^\varepsilon$. It is known for string graphs (and thus segment graphs) that in this case $tw(G) = \text{poly}(k^\varepsilon)n^{1/2}$ (corresponding to item 3 of the definition of nice). Thus, our goal is to reduce $|V(G)|$ to $\mathcal{O}(k^{2-\varepsilon'})$ for some ε' .

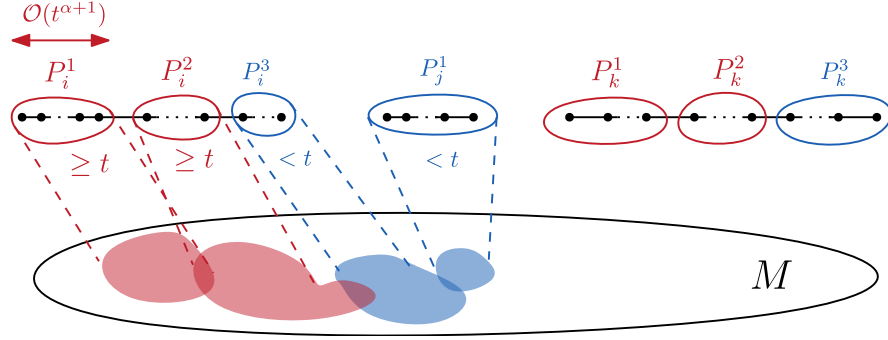
A first obvious rule is to iteratively contract edges of $G - M$ whose endpoints have no neighbors in M . This explains why we need the property of item 1 of the definition of nice.

Let us now explain how the property of item 2 (bounded tree neighborhood complexity, abbreviated bounded T-NC) allows to obtain the following “degree-related size property”: for any subpath P of a P_i , $|P| \leq \text{poly}(r)(d_M(P))^{\alpha+1}$. Define an independent set $\mathcal{T}$ of size $|P|/2$ by picking every second vertex in P . According to the definition of bounded T-NC (Definition 4) with $A = N_M(P)$, we get $|\{N_M(v), v \in \mathcal{T}\}| \leq f_1(r)(d_M(P))^\alpha$. Thus, if $|P| \geq x \cdot f_1(r)(d_M(P))^\alpha$, we found x vertices in P having the same neighborhood M' in M , and thus a vertex $u \in M'$ adjacent to x vertices in P . If x is large enough (about $d_M(P)$), it is always better to take any arbitrary vertex $u \in M'$. This leads to a kernelization rule (denoted (KR₅) in the full version): if there is a vertex $u \in M$ adjacent to approximately $d_M(P)$ vertices in a subpath P , delete u and decrease k by one. To sum it up, after applying this rule, for any subpath P of a P_i , if we have $d_M(P) = \text{poly}(r)$, then $|P| = \text{poly}(r)$.

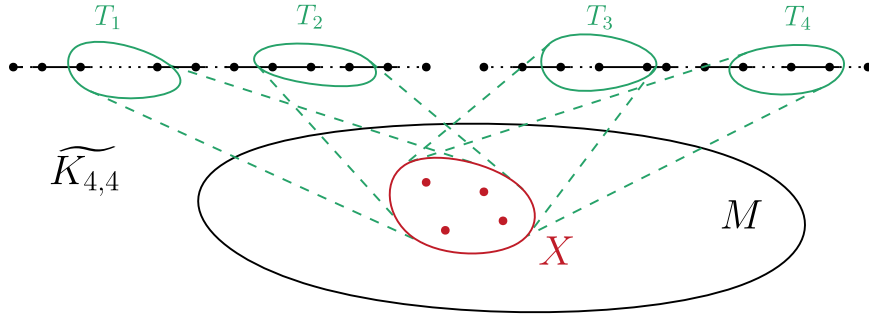
Before the next step we need to apply the following “large degree rule” (denoted (KR₃) in the full version): if there is a vertex v in a P_i such that $d_M(v) > t$, then add v to M . One can prove that by taking $t = 2d_r$, with $d_r = \text{poly}(r)$ the constant defined in item 4 of the definition of a nice class, M does not grow too much after applying this rule exhaustively: by denoting $A \subseteq M$ the set of vertices already in M before applying this rule, we always have $|A| = \text{poly}(r)|M \setminus A|$. This claim will be discussed in Subsection 2.3.

Observe that at this stage we may still have a large $K_{r,r}$ as minor, with for example graphs as in Figure 1 where no rule applies. It remains to define a crucial rule to destroy these $K_{r,r}$ minors. Let us now present a partitioning algorithm of the connected components in $G - M$. For any connected component P_i in $G - M$, we start (see Figure 2) from an endpoint of P_i and collect greedily vertices until we find a subpath P_i^1 such that $d_M(P_i^1) \geq t$, or that there is no more vertices in P_i . If $d_M(P_i^1) \geq t$, then restart a new path starting from the next vertex to create P_i^2 , and so on. This defines a partition $P_i = \bigcup_{\ell \in [x(P_i)]} (P_i^\ell)$, where $d_M(P_i^\ell) \geq t$ for any $\ell \in [1, x(P_i) - 1]$ and no lower bound for $d_M(P_i^{x(P_i)})$. As we applied the large degree rule, we also know that $d_M(P_i^\ell) \leq 2t$ for any $\ell \in [1, x(P_i)]$, because collecting at each step a new vertex in the path P_i^ℓ can increase $d_M(P_i^\ell)$ by at most t . This implies, using the degree-related size property introduced above, that $|P_i^\ell| \leq \text{poly}(r)$ for any $\ell \in [1, x(P_i)]$. Let us denote $\mathcal{T}^+$ the set of P_i^ℓ such that $d_M(P_i^\ell) \geq t$ the “large-degree subpaths”. Observe that the last considered subpath $P_i^{x(P_i)}$ of each connected component P_i (on the right of each path of $G - M$ in Figure 2) may have $d_M(P_i^{x(P_i)}) < t$ as there was no more vertices to complete it, hence it is not contained in $\mathcal{T}^+$. We denote $\mathcal{T}^-$ those remaining “small-degree subpaths”.

Let us now explain how the bounded T-NC property (Definition 4) allows to obtain the following “small number of large-degree subpaths” property. By removing half of the $\mathcal{T}^+$ s, we can get a set $\mathcal{T}^{+'}$ of non-adjacent trees (meaning with no edge between the T_i' s) such that $|\mathcal{T}^{+'}| \geq \frac{1}{2}|\mathcal{T}^+|$. We can then apply the T-NC property with $A = M$, $\mathcal{T} = \mathcal{T}^{+'}$ and $p = m = \text{poly}(r)$ to obtain $|\{N_M(T), T \in \mathcal{T}^{+'}\}| \leq \text{poly}(r)|M|$. Thus, if $|\mathcal{T}^{+'}| \geq x \cdot \text{poly}(r)|M|$, we found x large-degree subpaths (denoted T_i') having the same



■ **Figure 2** Example of partition where P_i is partitioned into $x(P_i) = 3$ subpaths, with P_i^1 and P_i^2 in $\mathcal{T}^+$ and $P_i^3 \in \mathcal{T}^-$.



■ **Figure 3** Example of $\widetilde{K}_{t,t}$ for $t = 4$. Here we have $X \subseteq N_M(T_i)$ for $1 \leq i \leq 4$.

neighborhood X' in M with $|X'| \geq t$. By choosing $x = t$ and considering $X \subseteq X'$ with $|X| = t$, we found a structure that we call a $\widetilde{K}_{t,t}$ (see Figure 3), formally defined as a pair $(X, (T_i)_{1 \leq i \leq t})$ where

- (P1) $X \subseteq M$ has size t ,
- (P2) $(T_i)_i$ a family of t vertex-disjoint non-adjacent subtrees (paths here) of $G - M$ such that for all $1 \leq i \leq t$, $X \subseteq N_M(T_i)$, and
- (P3) for any T_i , $|T_i| \leq \text{poly}(r)$.

Now, inspired by the “virtual branching rule” for a triangle bundle, we introduce a branching rule (denoted (BR₂) in the full version) that either deletes almost all vertices in X , or is adding to M a subset of $t - 1$ of the T'_i s. The complexity behind this rule is fine as in the second branch, the parameter virtually decreases by $t - 1$, and M grows by $(t - 1) \max_i \{|T_i|\} = \text{poly}(r)$. This explains how we deal with the $K_{t,t}$ -minors.

These rules do not suffice to bound $|V(G) \setminus M|$. Recall that any $P \in \mathcal{T}^+ \cup \mathcal{T}^-$ is such that $|P| \leq \text{poly}(r)$, and thus we only need to bound $|\mathcal{T}^+ \cup \mathcal{T}^-|$. As we cannot apply the previous rule, we know that the number of big paths is small: $|\mathcal{T}^+| \leq \text{poly}(r)|M|$. Now, to bound $|\mathcal{T}^-|$, observe that we can partition $\mathcal{T}^- = \mathcal{T}_1^- \cup \mathcal{T}_2^-$, where

- $\mathcal{T}_1^-$ is the set of small-degree paths P_i^ℓ for some $\ell > 1$ (belonging to the same path P_i than a large-degree path $P_i^{\ell-1}$).
- $\mathcal{T}_2^-$ is the set of small-degree paths which are entire connected components of $G - M$.

As $|\mathcal{T}_1^-| \leq |\mathcal{T}^+|$, it only remains to bound $|\mathcal{T}_2^-|$. Now, we can exploit once again the bounded T-NC property (Definition 4) to obtain that, if $|\mathcal{T}_2^-| \geq x \cdot \text{poly}(r)|M|$, then we can find x disjoint non-adjacent paths in $\mathcal{T}_2^-$ having the same neighborhood X in M . This case is different from the $\widetilde{K}_{t,t}$ case as X may be arbitrarily small (we only know that $|X| < t$), and

thus unlike $\widetilde{K_{t,t}}$ this does not decrease the parameter k by a large amount. However, in this case, paths of $\mathcal{T}_2^-$ are just connected components of $G - M$, and we use a last rule (denoted KR_4 in the full version) that identifies paths that can be safely removed. It can be shown that if $x+2$ paths have the same neighborhood in M , of size x , one of the paths is redundant. Hence after applying the rule exhaustively we can assume $|\mathcal{T}_2^-| < x \cdot \text{poly}(r)|M| = \text{poly}(r)|M|$.

This concludes the sketch of proof for this restricted setting where the connected components of $G - M$ are paths, as we obtain by taking ε small enough $|V(G) \setminus M| \leq \text{poly}(r)|M| \leq \text{poly}(k^\varepsilon)k = \mathcal{O}(k^{2-\varepsilon'})$ as required.

2.3 Challenges to lift the result from paths to trees

We now consider the real setting where given (G, k) where G is $K_{r,r}$ -free for $r = k^\varepsilon$, and given M a feedback vertex set of size at most $2k$, we want to reduce the graph to obtain $|V(G) \setminus M| = \mathcal{O}(k^{2-\varepsilon'})$. The approach still consists in partitioning $G - M$ in an appropriate way (called a t -uniform partition).

A first problem when trying to adapt the approach of Subsection 2.2 is the degree-related size property. Indeed, for trees we are now only able to obtain that for any subtree T of $G - M$, $|T| \leq \text{poly}(r)\mu(T)^{\mathcal{O}(\alpha)}$ where $\mu(T) = \max(d_M(T), b_{\overline{M}}(T))$ and $b_{\overline{M}}(T) = |\{v \in T, N(v) \not\subseteq M \cup T\}|$ is the size of the “border of T ”. Observe that $b_{\overline{M}}(P)$ is at most 2 for any subpath P of path P_i , whereas $b_{\overline{M}}(T)$ can only be bounded by $|T|$ for a subtree T . Informally, in the path case $|P|$ was only polynomially dependent on $d_M(P)$, and now $|T|$ is also polynomially depends on $b_{\overline{M}}(T)$.

A second problem is the large degree rule. Suppose that this rule no longer applies (meaning that for every $u \in V(G - M)$, we have $d_M(u) \leq t$), and suppose now that because of another rule a vertex $v \in V(G) \setminus M$ is added to M , denoting $M' = M \cup \{v\}$. Then this can create a new large degree vertex v' with $d_{M'}(v') > t$ (and so $d_M(v') = t$). Then v' would need to be added and the problem may arise again for another vertex v'' . This “cascading” can easily be prevented if $G - M$ is a forest of paths: it suffices to apply the rule a first time at the start of the algorithm, but with $t' = t - 2$. We then have for each $v \in V(G - M)$ the bound $d_M(v) \leq t - 2$, and we do not need to apply the rule again after as adding vertices to M may increase $d_M(v)$ by at most 2 ensuring the wanted bound $d_M(v) \leq t$ for $v \in V(G - M)$. However, in the case of a tree, we can have in $G - M$ a vertex of arbitrarily large degree, so we cannot apply the same solution. The problem is treated with the help of a technical lemma which ensures that throughout the execution of the algorithm we keep $|M| = \text{poly}(r)k$.

Finally, a third problem is the definition of the partition. As in the case of paths we want to partition $G - M$ into a “ t -uniform partition” $\mathcal{T}$, where in particular we have $\mathcal{T} = \mathcal{T}^+ \cup \mathcal{T}^-$, and for any $T \in \mathcal{T}$, $d_M(T) \leq 2t$ and $|T| \leq \text{poly}(r)\mu(T)^{\mathcal{O}(\alpha)}$. The greedy approach presented for the case of paths is now more involved, as we have to cut each tree of $G - M$ into subtrees that have small border $b_{\overline{M}}(T)$, as otherwise the previous bound $|T| \leq \text{poly}(r)\mu(T)^{\mathcal{O}(\alpha)}$ becomes useless when $\mu(T)$ is too large.

This partitioning procedure can either:

- Fail and find a subtree T with $|T| > \text{poly}(r)\mu(T)^{C\alpha}$, for some constant C , implying that our degree-related size rule can be applied.
- Fail and find too many subtree $T_i \in \mathcal{T}^+$ with large degree, implying that we found a $\widetilde{K_{t,t}}$, and that our $\widetilde{K_{t,t}}$ rule can be applied.
- Produce a t -uniform partition with $|\mathcal{T}^+| \leq \text{poly}(r)|M|$.

In the third case, we either find another way to apply one more time a reduction rule, or prove that $|V(G) \setminus M| = \mathcal{O}(k^{2-\varepsilon'})$.

2.4 An important tool to prove that an intersection graph class is nice

We believe that one important application of our main theorem is the class of s -string graphs (recall that segment graphs are a subclass of 1-string graphs). To prove that s -string graphs is a nice class (and thus captured by our result), one needs to prove that s -string graphs have a bounded tree neighborhood complexity. This property is not straightforward, and we would like to point out the following theorem which seems useful to obtain results related to neighborhood complexity.

► **Theorem 8** ([16]). *Given $\mathcal{F}$ a family of pseudo-disks and $\mathcal{B}$ a finite family of pseudo-disks, consider the hypergraph $H(\mathcal{B}, \mathcal{F})$ whose vertices are the pseudo-disks in $\mathcal{B}$ and the edges are all subsets of $\mathcal{B}$ of the form $\{D \in \mathcal{B}, D \cap S \neq \emptyset\}$, with $S \in \mathcal{F}$. Then the number of edges of cardinality at most $k \geq 1$ in $H(\mathcal{B}, \mathcal{F})$ is $\mathcal{O}(|\mathcal{B}|k^3)$.*

A very important aspect in this result is that it is not required that $\mathcal{F} \cup \mathcal{B}$ is a family of pseudo-disks, and thus pseudo-disks of $\mathcal{B}$ may “cross” pseudo disks of $\mathcal{F}$. This is indeed the case in our application where in particular we associate to each tree in $G - M$ a pseudo-disk in $\mathcal{B}$, and this pseudo disk may cross pseudo-disks associated to vertices of $\mathcal{M}$ (see proof of Lemma 11).

3 Applications

In this section, we prove that s -string graphs and pseudo-disk graphs are nice graph classes for some parameters. We recall that a system of pseudo-disks is a collection of regions in the plane homeomorphic to a disk such that any two of them share at most 2 points of their boundary. Similar arguments are used for both considered classes, but in the case of pseudo-disks the arguments are a bit simpler, we then consider this class in first.

3.1 Application to pseudo-disk graphs

In this section, we prove that the class of pseudo-disk graphs is nice, and so by our main theorem it admits a robust subexponential FPT algorithm for FVS. Note that for this graph class, the existence of such algorithm was already known [4].

► **Lemma 9.** *There exists a constant c such that the class of pseudo-disk graphs has bounded tree neighborhood complexity with parameters $\alpha = 4$, $f_1(r) = c$ and $f_2(r, p, m) = cp^3$.*

Proof. Let $r \geq 2$, G be a $K_{r,r}$ -free pseudo-disk graph, $(\mathcal{D}_v)_{v \in V(G)}$ a representation of G as pseudo-disks, $A \subseteq V(G)$, and $\mathcal{T}$ a family of disjoint non-adjacent trees of $G - A$. We would like to apply Theorem 8 on the set A and the family of subsets of the plane obtained by taking for each tree $T \in \mathcal{T}$ the union $\mathcal{D}_T = \cup_{v \in T} \mathcal{D}_v$. Because we are considering pseudo-disks and trees, this union is homeomorphic to a disk. Moreover as the trees are non-adjacent the obtained unions do not intersect each other, and so are a pseudo-disk system trivially. Now applying Theorem 8 with the families $(\mathcal{D}_v)_{v \in A}$ and $(\mathcal{D}_T)_{T \in \mathcal{T}}$ directly gives that there exists a constant c_2 such that if for all $T \in \mathcal{T}$ we have $d_A(T) \leq p$ then $|\{N_A(T), T \in \mathcal{T}\}| \leq c_2 p^3 |A|$ as wanted for the second bound. Observing that we can take $p = |A|$ as an upper bound for $d_A(T)$ with $T \in \mathcal{T}$ gives the first wanted bound. ◀

► **Lemma 10.** *There exist constants c_1, c_2, c_3, c_4 such that the class of pseudo-disk graphs is a nice class with parameters $\alpha = 4$, $f_1(r) = c_1$, $f_2(r, p, m) = c_2 p^3$, $\delta = \frac{1}{2}$, $f_r = c_3 \sqrt{r \log r}$ and $d_r = c_4 r \log r$.*

Proof. Most of the conditions were already stated with Theorem 2, Theorem 3 and Lemma 9. The only property that remains to check is that in a pseudo-disk graph G , identifying adjacent degree-two vertices u and v obtaining a new vertex w preserves being a pseudo-disk graph. We may assume that the vertices do not have a common neighbor as otherwise we obtain an induced subgraph of G which is trivially a pseudo-disk graph. By considering $(\mathcal{D}_q)_{q \in V(G)}$ a pseudo-disk representation of G , we can take $\mathcal{D}_w = \mathcal{D}_u \cup \mathcal{D}_v$ as it is homeomorphic to a disk and its boundary will cross the second neighbor of u (respectively v) as many time than the boundary of $\mathcal{D}_u$ (respectively $\mathcal{D}_v$). ◀

And so applying our main theorem we obtain:

► **Corollary 7.** *There exists $\eta < 1$, such that there is a robust parameterized subexponential algorithm solving FVS in time $2^{\mathcal{O}(k^\eta)} n^{\mathcal{O}(1)}$ for n -vertex pseudo-disk graphs.*

Note that our main theorem gives that it suffices to take $\eta > \frac{44}{45}$ for the result to hold. Sharper results already exist in the literature for pseudo-disk graphs, by generalizing the robust algorithm of [21] dealing with disk graphs to pseudo-disk graphs, or by using the representation as pseudo-disks for obtaining an even better running time. See [6], the full version of [4] for both methods.

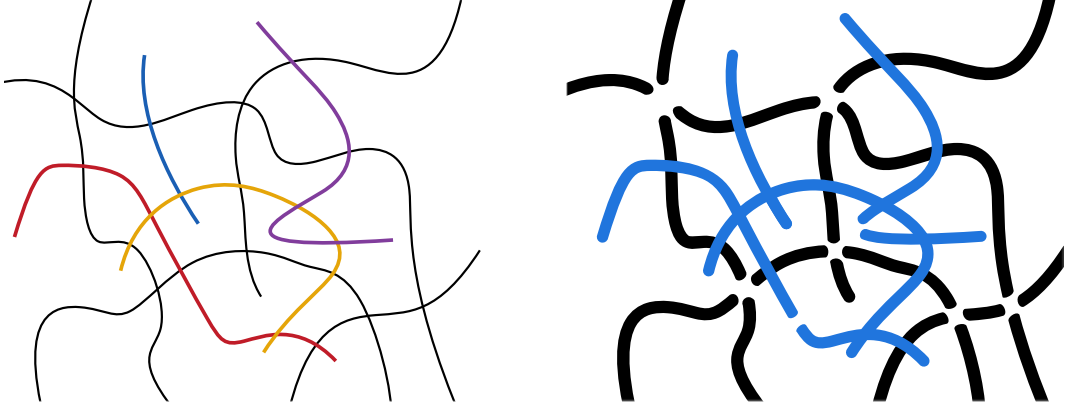
3.2 Application to s -string graphs

We now show how to adapt the arguments from the previous section to the case of s -strings.

► **Lemma 11.** *There exist constants c_1, c_2 such that the class of s -string graphs has bounded tree neighborhood complexity with parameter $\alpha = 4$, $f_1(r) = c_1 s^4 r \log r$ and $f_2(r, p, m) = c_2 (sr \log r)^4 (p + m)^3$.*

Proof. The proof is very similar to the proof of Lemma 9 with some additional tweaks. Let G be a $K_{r,r}$ -free s -string graph for some $r \geq 2$ and $(\mathcal{S}_v)_{v \in V(G)}$ be an s -string representation of G . For $T \in \mathcal{T}$, again we consider the region $\mathcal{D}_T = \bigcup_{v \in V(T)} \mathcal{S}_v$. In this case however, even so T is a tree, the complement of this region is not necessarily connected (for example if two strings of T intersect on several points). In order to obtain a region of the plane homeomorphic to a disk, we modify $\mathcal{D}_T$ by cutting small sections where no intersections take place (not even with a string of A) and obtain a new set $\mathcal{D}'_T \subseteq \mathcal{D}_T$ that is connected and such that there is exactly one arc-connected region in $\mathbb{R}^2 \setminus \mathcal{D}'_T$ and which intersects the same strings in A as $\mathcal{D}_T$ (see Figure 4). Moreover we add a small thickness to $\mathcal{D}'_T$ while preserving the wanted property about the number of regions in $\mathbb{R}^2 \setminus \mathcal{D}'_T$ in order to obtain a geometric shape homeomorphic to a disk. Observe that as there are no edges between vertices of distinct members of $\mathcal{T}$, by taking the thickness small enough the connected sets of the form $\mathcal{D}'_T$ (for $T \in \mathcal{T}$) are not crossing and so they trivially form a system of pseudo-disks.

Now we would like to construct another system of pseudo-disks representing the vertices of A . We do this by cutting the strings in $(\mathcal{S}_u)_{u \in A}$ whenever two of them cross each other. Observe that by Theorem 3 there exists a constant c such that by denoting $d_r = cr \log r$ we have at most $d_r |A|$ intersecting pairs, and remember that each pair intersects each other at most s times as we consider a s -string representation. So we have at most $sd_r |A|$ cuts which gives an upper bound of $(sd_r + 1)|A|$ for the number of sections of strings between intersection points. We denote C the set of such sections. Again we define a family $(\mathcal{R}_v)_{v \in C}$ by considering the sections (which are disjoint by construction) and giving them some thickness, while preventing the sections to intersect each other. This is clearly a system of pseudo-disks as its elements are disjoint. By Theorem 8, the number of distinct sets



■ **Figure 4** Transformation of a union of s -strings to a region homeomorphic to a disk, while keeping the same neighborhood in A . The strings of A are depicted in black.

$N_C(\mathcal{D}'_T) = \{v \in C, \mathcal{D}'_T \text{ intersects } \mathcal{R}_v\}$, for $T \in \mathcal{T}$, is $\mathcal{O}(x^3|C|)$ where x is the maximum, over every $T \in \mathcal{T}$, of the number of pseudo-disks in $(\mathcal{R}_v)_{v \in C}$ crossed by $\mathcal{D}'_T$. This upper bound gives the second result of this lemma by observing that if a pair of pseudo-disks $\mathcal{D}'_{T_1}, \mathcal{D}'_{T_2}$ intersect exactly the same set of $\mathcal{R}_v$, then $N_A(T_1) = N_A(T_2)$. We obtain the first result of the lemma by using the trivial bound $x \leq |C|$. ◀

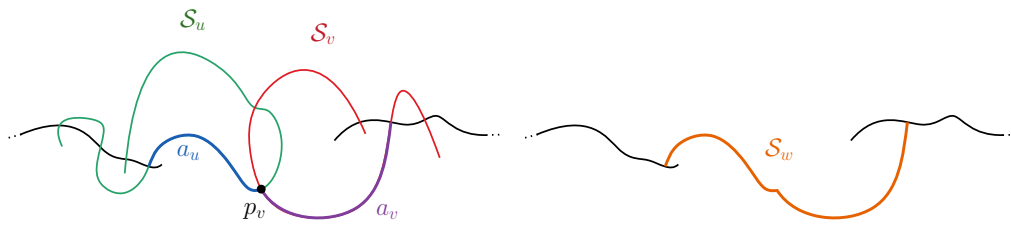
► **Lemma 12.** *There exist constants c_1, c_2, c_3, c_4 such that for every $s \geq 1$ the class of s -string graphs is a nice class with parameters $\alpha = 4$, $f_1(r) = c_1 s^4 r \log r$, $f_2(r, p, m) = c_2 (sr \log r)^4 (p + m)^3$, $\delta = \frac{1}{2}$, $f_r = c_3 \sqrt{r \log r}$ and $d_r = c_4 r \log r$.*

Proof. Again most of the conditions were already stated with Theorem 2, Theorem 3 and Lemma 11. The only remaining property to prove is that given an s -string graph G , identifying adjacent degree-two vertices u and v preserves being a s -string graph. We may assume that u and v do not have a common neighbor as otherwise the obtained graph is an induced subgraph of G and so is a s -string graph. We do this by replacing in a s -string representation $(\mathcal{S}_q)_{q \in V(G)}$ of G the strings $\mathcal{S}_u$ and $\mathcal{S}_v$ with a new string $\mathcal{S}_w$ while preserving the global property of being an s -string family. $\mathcal{S}_u$ contains a Jordan arc a_u linking its two neighbors, without being intersected by any string in its interior. Let denote p_v the extremity of a_u in $\mathcal{S}_v$. Then $\mathcal{S}_v$ contains a Jordan arc a_v linking p_v to the string representing its second neighbor, without being intersected by any other string than $\mathcal{S}_u$. The string $\mathcal{S}_w$ is then obtained by taking the union of a_u and a_v . This construction ensures that the number of crossings between two strings other than $\mathcal{S}_u$ and $\mathcal{S}_v$ of the initial representation remain the same. We now have to bound the number of crossing of $\mathcal{S}_w$ with others strings. Recalling that u and v do not share a common neighbor, and that by construction w has 2 crossings (one for each distinct neighbor), w does not have more than 1 crossings with another string. So after replacing $\mathcal{S}_u$ and $\mathcal{S}_v$ by $\mathcal{S}_w$ we still have a s -string family. ◀

And so applying our main theorem we obtain:

► **Corollary 6.** *There exists $\eta < 1$, such that for all s there is a robust parameterized subexponential algorithm solving FVS in time $2^{\tilde{\mathcal{O}}(s^{\mathcal{O}(1)} k^\eta)} n^{\mathcal{O}(1)}$ for n -vertex s -string graphs.*

More precisely, we can take $\eta = \frac{52}{53}$.



■ **Figure 5** Illustration of the construction used in the proof of Lemma 12 for contracting an edge between adjacent degree-two vertices without a common neighbor.

4 Conclusion

In this paper we gave general sufficient conditions for the existence of subexponential parameterized algorithms for the FEEDBACK VERTEX SET problem. Our main theorem unifies the previously known results on several classes of graphs such as planar graphs, map graphs⁷, unit-disk graphs, disk graphs, or more generally pseudo-disk graphs⁸, and string graphs of bounded edge-degree. It also applies to classes where such algorithms were not known to exist, notably intersection graphs of thin objects such as segment graphs and more generally s -string graphs.

However, we have so far no evidence that our concept of nice class could be an answer to Question 1. So a natural direction for future research would be to investigate more general graph classes than those listed above or to discover new classes that fit in our framework. There are two natural candidates:

- Intersection graphs of objects in higher dimensions. However, as proved in [15], under ETH intersection graphs of unit balls in $\mathbb{R}^3$ do not admit a subexponential parameterized algorithm for FVS.
- Graph classes excluding an induced minor.⁹ This is more general than string graphs (which exclude a subdivided K_5 as an induced minor). In such classes, Korhonen and Lokshtanov [17] recently provided an algorithm solving FVS in time $2^{\mathcal{O}_H(n^{2/3} \log n)}$, that is, subexponential in the input size n .

Less general than the previous item is the class of string graphs, which is nevertheless the most general class of intersection graphs of (arc-connected) objects in the Euclidean plane. At the time of the writing, it is not excluded that it could be a nice class. So far, we are still missing the property about the bounded tree neighborhood complexity. Note that in our proof for s -strings, the bound on the number s of crossing is only used to prove that the class satisfies this property of bounded tree neighborhood complexity. Obtaining a similar result without using the bound on the number of crossings would thus yield the desired generalization. But an even simpler way could be to prove that the general case of string graphs can be reduced to the case of s -strings graphs for some “small” s function of the problem parameter k . A first idea would be to prove that $K_{t,t}$ -free string graphs are t^c -string graphs for some constant c , however this fails as there are string graphs with n vertices, and so which are $K_{n,n}$ -free, that require 2^{cn} crossings for some constant c [19].

⁷ Map graphs are intersection graphs of interior-disjoint regions of $\mathbb{R}^2$ homeomorphic to disks.

⁸ Pseudo-disk graphs are intersection graphs of a collection of regions in the plane homeomorphic to a disk such that any two of them share at most 2 points of their boundary

⁹ A graph H is said to be an *induced minor* of a graph G if it can be obtained from G by deleting vertices and contracting edges. Otherwise, G is said to *exclude* H as induced minor or to be *H -induced minor free*.

References

- 1 Shinwoo An, Kyungjin Cho, and Eunjin Oh. Faster algorithms for cycle hitting problems on disk graphs. In *Algorithms and Data Structures: 18th International Symposium, WADS 2023, Montreal, QC, Canada, July 31 – August 2, 2023, Proceedings*, pages 29–42, Berlin, Heidelberg, 2023. Springer-Verlag. doi:10.1007/978-3-031-38906-1_3.
- 2 Shinwoo An and Eunjin Oh. Feedback Vertex Set on Geometric Intersection Graphs. In Hee-Kap Ahn and Kunihiko Sadakane, editors, *32nd International Symposium on Algorithms and Computation (ISAAC 2021)*, volume 212 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 47:1–47:12, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ISAAC.2021.47.
- 3 Julien Baste and Dimitrios M Thilikos. Contraction bidimensionality of geometric intersection graphs. *Algorithmica*, 84(2):510–531, 2022. doi:10.1007/S00453-021-00912-W.
- 4 Gaétan Berthe, Marin Bougeret, Daniel Gonçalves, and Jean-Florent Raymond. Feedback Vertex Set for pseudo-disk graphs in subexponential FPT time. In Daniel Král and Martin Milanič, editors, *Graph-Theoretic Concepts in Computer Science*, pages 65–78, Cham, 2025. Springer Nature Switzerland.
- 5 Gaétan Berthe, Marin Bougeret, Daniel Gonçalves, and Jean-Florent Raymond. Subexponential Algorithms in Geometric Graphs via the Subquadratic Grid Minor Property: The Role of Local Radius. In Hans L. Bodlaender, editor, *19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*, volume 294 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:18, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SWAT.2024.11.
- 6 Gaétan Berthe, Marin Bougeret, Daniel Gonçalves, and Jean-Florent Raymond. Feedback vertex set for pseudo-disk graphs in subexponential fpt time, 2024. doi:10.48550/arXiv.2410.23878.
- 7 Gaétan Berthe, Marin Bougeret, Daniel Gonçalves, and Jean-Florent Raymond. Pushing the frontiers of subexponential fpt time for feedback vertex set, 2025. arXiv:2504.17708.
- 8 Édouard Bonnet, Florent Foucaud, Tuomo Lehtilä, and Aline Parreau. Neighbourhood complexity of graphs of bounded twin-width. *European Journal of Combinatorics*, 115:103772, 2024. doi:10.1016/J.EJC.2023.103772.
- 9 Édouard Bonnet and Paweł Rzażewski. Optimality program in segment and string graphs. *Algorithmica*, 81:3047–3073, 2019. doi:10.1007/S00453-019-00568-7.
- 10 Édouard Bonnet and Paweł Rzażewski. An 11/6-approximation algorithm for vertex cover on string graphs, 2024. doi:10.48550/arXiv.2409.18820.
- 11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 1st edition, 2015.
- 12 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for exponential-time-hypothesis-tight algorithms and lower bounds in geometric intersection graphs. *SIAM Journal on Computing*, 49(6):1291–1331, 2020. doi:10.1137/20M1320870.
- 13 Reinhard Diestel. Graph theory 3rd ed. *Graduate texts in mathematics*, 173(33):12, 2005.
- 14 Zdeněk Dvořák and Sergey Norin. Treewidth of graphs with balanced separations. *Journal of Combinatorial Theory, Series B*, 137:137–144, 2019. doi:10.1016/j.jctb.2018.12.007.
- 15 Fedor V Fomin, Daniel Lokshtanov, and Saket Saurabh. Excluded grid minors and efficient polynomial-time approximation schemes. *Journal of the ACM (JACM)*, 65(2):1–44, 2018. doi:10.1145/3154833.
- 16 Balázs Keszegh. Coloring intersection hypergraphs of pseudo-disks. *Discret. Comput. Geom.*, 64(3):942–964, 2020. doi:10.1007/S00454-019-00142-6.
- 17 Tuukka Korhonen and Daniel Lokshtanov. Induced-minor-free graphs: Separator theorem, subexponential algorithms, and improved hardness of recognition. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2024.

- 18 Jan Kratochvíl and Jiří Matoušek. Intersection graphs of segments. *Journal of Combinatorial Theory, Series B*, 62(2):289–315, 1994. doi:10.1006/JCTB.1994.1071.
- 19 Jan Kratochvíl and Jiří Matoušek. String graphs requiring exponential representations. *Journal of Combinatorial Theory, Series B*, 53(1):1–4, 1991. doi:10.1016/0095-8956(91)90050-T.
- 20 James R. Lee. Separators in Region Intersection Graphs. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:8, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2017.1.
- 21 Daniel Lokshantov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. Subexponential parameterized algorithms on disk graphs (extended abstract). In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2005–2031. SIAM, 2022. doi:10.1137/1.9781611977073.80.
- 22 Daniel Lokshantov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. A framework for approximation schemes on disk graphs. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2228–2241. SIAM, 2023. doi:10.1137/1.9781611977554.CH84.
- 23 Daniel Lokshantov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. Subexponential parameterized algorithms for hitting subgraphs. *arXiv preprint arXiv:2409.04786*, 2024. doi:10.48550/arXiv.2409.04786.
- 24 Jiří Matoušek. Near-optimal separators in string graphs. *Comb. Probab. Comput.*, 23(1):135–139, 2014. doi:10.1017/S0963548313000400.
- 25 Jiří Matoušek. String graphs and separators. In *Geometry, structure and randomness in combinatorics*, pages 61–97. Springer, 2014. doi:10.1007/978-88-7642-525-7_5.
- 26 Felix Reidl, Fernando Sánchez Villaamil, and Konstantinos Stavropoulos. Characterising bounded expansion by neighbourhood complexity. *European Journal of Combinatorics*, 75:152–168, 2019. doi:10.1016/J.EJC.2018.08.001.