Parameterized Max Min Feedback Vertex Set

Michael Lampis

Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016, Paris, France

Nikolaos Melissinos

Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Czech Republic

Manolis Vasilakis

Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016, Paris, France

Abstract

Given a graph G and an integer k, Max Min FVS asks whether there exists a minimal set of vertices of size at least k whose deletion destroys all cycles. We present several results that improve upon the state of the art of the parameterized complexity of this problem with respect to both structural and natural parameters.

Using standard DP techniques, we first present an algorithm of time $\mathrm{tw}^{O(\mathrm{tw})}n^{O(1)}$, significantly generalizing a recent algorithm of Gaikwad et al. of time $\mathrm{vc}^{O(\mathrm{vc})}n^{O(1)}$, where tw, vc denote the input graph's treewidth and vertex cover respectively. Subsequently, we show that both of these algorithms are essentially optimal, since a $\mathrm{vc}^{o(\mathrm{vc})}n^{O(1)}$ algorithm would refute the ETH.

With respect to the natural parameter k, the aforementioned recent work by Gaikwad et al. claimed an FPT branching algorithm with complexity $10^k n^{O(1)}$. We point out that this algorithm is incorrect and present a branching algorithm of complexity $9.34^k n^{O(1)}$.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases ETH, Feedback vertex set, Parameterized algorithms, Treewidth

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.62

Related Version *Full Version*: https://arxiv.org/abs/2302.09604

Funding This work is partially supported by ANR projects ANR-21-CE48-0022 (S-EX-AP-PE-AL) and ANR-18-CE40-0025-01 (ASSK).
Nikolaos Melissinos: Supported by the CTU Global postdoc fellowship program.

Acknowledgements Work primarily conducted while Nikolaos Melissinos was affiliated with Université Paris-Dauphine.

1 Introduction

We consider a MaxMin version of the well-studied feedback vertex set problem where, given a graph $G = (V, E)$ and a target size k, we are asked to find a set of vertices S with the following properties: (i) every cycle of G contains a vertex of S, that is, S is a feedback vertex set (ii) no proper subset of S is a feedback vertex set, that is, S is minimal (iii) $|S| \geq k$. Although much less studied than its minimization cousin, Max Min FVS has recently attracted attention in the literature as part of a broader study of MaxMin versions of standard problems, such as Max Min Vertex Cover and Upper Dominating Set. The main motivation of this line of research is the search for a deeper understanding of the performance of simple greedy algorithms: given an input, we would like to compute what is the worst possible solution that would still not be improvable by a simple heuristic, such as removing redundant vertices. Nevertheless, over recent years MaxMin problems have been found to possess an interesting combinatorial structure of their own and have now become an object of more widespread study (we survey some such results below).

It is not surprising that Max Min FVS is known to be NP-complete and is in fact significantly harder than Minimum FVS in most respects, such as its approximability or its amenability to algorithms solving special cases. Given the problem's hardness, in this paper we focus on the parameterized complexity of Max Min FVS, since parameterized complexity is one of the main tools for dealing with computational intractability[1]. We consider two types of parameterizations: the natural parameter k; and the parameterization by structural width measures, such as treewidth. In order to place our results into perspective, we first recall the current state of the art.

Previous work. Max Min FVS was first shown to be NP-complete even on graphs of maximum degree 9 by Mishra and Sikdar [32]. This was subsequently improved to NP-completeness for graphs of maximum degree 6 by Dublois et al. [20], who also present an approximation algorithm with ratio $n^{2/3}$ and proved that this is optimal unless P=NP. A consequence of the polynomial time approximation algorithm of [20] was the existence of a kernel of order $O(k^3)$, which implied that the problem is fixed-parameter tractable with respect to the natural parameter k. Some evidence that this kernel size may be optimal was later given by [2]. We note also that the problem can easily be seen to be FPT parameterized by treewidth (indeed even by clique-width) as the property that a set is a minimal feedback vertex set is MSO_1-expressible, so standard algorithmic meta-theorems apply.

Given the above, the state of the art until recently was that this problem was known to be FPT for the two most well-studied parameterizations (by k and by treewidth), but concrete FPT algorithms were missing. An attempt to advance this state of the art and systematically study the parameterized complexity of the problem was recently undertaken by Gaikwad et al. [23], who presented exact algorithms for this problem running in time $10^k n^{O(1)}$ and $\mathrm{vc}^{O(\mathrm{vc})} n^{O(1)}$, where vc is the input graph's vertex cover, which is known to be a (much) more restrictive parameter than treewidth. Leveraging the latter algorithm, [23] also present an FPT approximation scheme which can $(1-\varepsilon)$-approximate the problem in time $2^{O(\mathrm{vc}/\varepsilon)} n^{O(1)}$, that is, single-exponential time with respect to vc.

Our contribution. We begin our work by considering Max Min FVS parameterized by the most standard structural parameter, treewidth. We observe that, using standard DP techniques, we can obtain an algorithm running in time $\mathrm{tw}^{O(\mathrm{tw})} n^{O(1)}$, that is, slightly super-exponential with respect to treewidth. Note that this slightly super-exponential running time is already present in the $\mathrm{vc}^{O(\mathrm{vc})} n^{O(1)}$ algorithm of [23], despite the fact that vertex cover is a much more severely restricted parameter. Hence, our algorithm generalizes the algorithm of [23] without a significant sacrifice in the running time.

Despite the above, our main contribution with respect to structural parameters is not our algorithm for parameter treewidth, but an answer to a question that is naturally posed given the above: can the super-exponential dependence present in both our algorithm and the algorithm of [23] be avoided, that is, can we obtain a $2^{O(\mathrm{tw})} n^{O(1)}$ algorithm? We show that this is likely impossible, as the existence of an algorithm running in time $\mathrm{vc}^{o(\mathrm{vc})} n^{O(1)}$ is ruled out by the ETH (and hence also the existence of a $\mathrm{tw}^{o(\mathrm{tw})} n^{O(1)}$ algorithm). This result is likely to be of wider interest to the parameterized complexity community, where one of the most exciting developments of the last fifteen years has arguably been the development of the Cut&Count technique (and its variations). One of the crowning achievements of this

[1] Throughout the paper we assume that the reader is familiar with the basics of parameterized complexity, as given in standard textbooks [16].

technique is the design of single-exponential algorithms for connectivity problems – indeed an algorithm running in time $3^{\mathrm{tw}}n$ for Minimum FVS is given in [17]. It has therefore been of much interest to understand which connectivity problems admit single-exponential algorithms using such techniques (see e.g. [7] and the references within). Curiously, even though several cousins of Minimum Feedback Vertex Set have been considered in this context (such as Subset Feedback Vertex Set and Restricted Edge-Subset Feedback Edge Set), for Max Min FVS, which is arguably a very natural variant, it was not known whether a single-exponential algorithm for parameter treewidth is possible. Our work thus adds to the literature a natural connectivity problem where Cut&Count can provably not be applied (under standard assumptions). Interestingly, our lower bound even applies to the case of vertex cover, which is rare, as most problems tend to become rather easy under this very restrictive parameter.

We then move on to consider the parameterization of the problem by k, the size of the sought solution. Observe that a $k^{O(k)}n^{O(1)}$ algorithm can easily be obtained by the results sketched above and a simple win/win argument: start with any minimal feedback vertex set S of the given graph G: if $|S| \geq k$ we are done; if not, then $\mathrm{tw}(G) \leq k$ and we can solve the problem using the algorithm for treewidth. It is therefore only interesting to consider algorithms with a single-exponential dependence on k. Such an algorithm, with complexity $10^k n^{O(1)}$, was claimed by [23]. Unfortunately, as we explain in detail in Section 5, this algorithm contains a significant flaw[2].

Our contribution is to present a corrected version of the algorithm of [23], which also achieves a slightly better running time of $9.34^k n^{O(1)}$, compared to the $10^k n^{O(1)}$ of the (flawed) algorithm of [23]. Our algorithm follows the same general strategy of [23], branching and placing vertices in the forest or the feedback vertex set. However, we have to rely on a more sophisticated measure of progress, because simply counting the size of the selected set is not sufficient. We therefore measure our progress towards a restricted special case we identify, namely the case where the undecided part of the graph induces a linear forest. Though this special case sounds tantalizingly simple, we show that the problem is still NP-complete under this restriction, but obtaining an FPT algorithm is much easier. We then plug in our algorithm to a more involved branching procedure which aims to either reduce instances into this special case, or output a certifiable minimal feedback vertex set of the desired size.

Finally, motivated by the above we note that a blocking point in the design of algorithms for Max Min FVS seems to be the difficulty of the extension problem: given a set S_0, decide if a minimal fvs S that extends S_0 exists. As mentioned, Casel et al. [13] showed that this problem is W[1]-hard parameterized by $|S_0|$. Intriguingly, however, it is not even known if this problem is in XP, that is, whether it is solvable in polynomial time for fixed k. We show that this is perhaps not surprising, as obtaining a polynomial time algorithm in this case would imply the existence of a polynomial time algorithm for the notorious k-in-a-Tree problem: given k terminals in a graph, find an induced tree that contains them. Since this problem was solved for $k = 3$ in a breakthrough by Chudnovsky and Seymour [15], the complexity for fixed $k \geq 4$ has remained a big open problem (for example [29] states that "Solving it in polynomial time for constant k would be a huge result"). It is therefore perhaps not surprising that obtaining an XP algorithm for the extension problem for minimal feedback vertex sets of fixed size is challenging, since such an algorithm would settle another long-standing problem.

[2] Saket Saurabh, one of the authors of [23], confirmed so via private communication with Michael Lampis.

Other relevant work. As mentioned, Max Min FVS is an example of a wider class of MaxMin problems which have recently attracted much attention in the literature, among the most well-studied of which are Maximum Minimal Vertex Cover [2, 11, 12, 34] and Upper Dominating Set (which is the standard name for Maximum Minimal Dominating Set) [1, 3, 5, 21]. Besides these problems, MaxMin or MinMax versions of cut and separations problems [19, 26, 30], knapsack problems [22, 24], matching problems [14], and coloring problems [6] have also been studied.

The question of which connectivity problems admit single-exponential algorithms parameterized by treewidth has been well-studied over the last decade. As mentioned, the main breakthrough was the discovery of the Cut&Count technique [16], which gave randomized $2^{O(\mathrm{tw})}n^{O(1)}$ algorithms for many such problems, such as Steiner Tree, Hamiltonicity, Connected Dominating Set and others. Follow-up work also provided deterministic algorithms with complexity $2^{O(\mathrm{tw})}n^{O(1)}$ [8]. It is important to note that the discovery of these techniques was considered a surprise at the time, as the conventional wisdom was that connectivity problems probably require $\mathrm{tw}^{O(\mathrm{tw})}$ time to be solved [31]. Naturally, the topic was taken up with much excitement, in an attempt to discover the limits of such techniques, including problems for which they cannot work. In this vein, [33] gave a meta-theorem capturing many tractable problems, and also an example problem that cannot be solved in time $2^{o(\mathrm{tw}^2)}n^{O(1)}$ under the ETH. Several other examples of connectivity problems which require slightly super-exponential time parameterized by treewidth are now known [4, 27], with the most relevant to our work being the feedback vertex set variants studied in [7, 10], as well as the digraph version of the minimum feedback vertex set problem (parameterized by the treewidth of the underlying graph) [9]. The results of our paper seem to confirm the intuition that the Cut&Count technique is rather fragile when applied to feedback vertex set problems, since in many variations or generalizations of this problem, a super-exponential dependence on treewidth is inevitable (assuming the ETH).

2 Preliminaries

Throughout the paper, we use standard graph notation [18]. Moreover, for vertex $u \in V(G)$, let $\deg_X(u)$ denote its degree in $G[X \cup \{u\}]$, where $X \subseteq V(G)$. A *multigraph* G is a graph which is permitted to have multiple edges with the same end nodes, thus, two vertices may be connected by more than one edge. Given a (multi)graph G, where $e = \{u, v\} \in E(G)$ is a not necessarily unique edge connecting distinct vertices u and v, the *contraction* of e results in a new graph G' such that $V(G') = (V(G) \setminus \{u, v\}) \cup \{w\}$, while for each edge $\{u, x\}$ or $\{v, x\}$ in $E(G)$, there exists an edge $\{w, x\}$ in $E(G')$. Any edge $e \in E(G)$ not incident to u, v also belongs to $E(G')$. If u and v were additionally connected by an edge apart from e, then w has a self loop.

For $i \in \mathbb{N}$, $[i]$ denotes the set $\{1, \dots, i\}$. A feedback vertex set S of G is minimal if and only if $\forall s \in S$, $G[(V(G) \setminus S) \cup \{s\}]$ contains a cycle, namely a *private cycle* of s [21]. Lastly, we make use of a weaker version of ETH, which states that 3-SAT cannot be determined in time $2^{o(n)}$, where n denotes the number of the variables [28].

Finally, note that the proofs of all lemmas and theorems marked with ($\star$) are present in the full version of the paper.

3 Treewidth Algorithm

Here we will present an algorithm for Max Min FVS parameterized by the treewidth of the input graph, arguably the most well studied structural parameter. As a corollary of the lower bound established in Section 4, it follows that the running time of the algorithm is essentially optimal under the ETH.

▶ **Theorem 1.** (⋆) *Given an instance $\mathcal{I} = (G, k)$ of* Max Min FVS*, as well as a nice tree decomposition of G of width* tw*, there exists an algorithm that decides $\mathcal{I}$ in time $\mathrm{tw}^{O(\mathrm{tw})}n^{O(1)}$.*

Proof sketch. The main idea lies on performing standard dynamic programming on the nodes of the nice tree decomposition. To this end, for each node, we will consider all the partial solutions, corresponding to (not necessarily minimal) feedback vertex sets of the subgraph induced by the vertices of the nodes of the corresponding subtree of the tree decomposition. We will try to extend such a feedback vertex set to a minimal feedback vertex set of G, that respects the partial solution. For each partial solution, it is imperative to identify, apart from the vertices of the bag that belong to the feedback vertex set, the connectivity of the rest of the vertices in the potential final forest. In order to do so, we consider a coloring indicating that, same colored vertices of the forest of the partial solution, should be in the same connected component of the potential final forest. Moreover, we keep track of which vertices of the forest of the partial solution are connected via paths containing forgotten vertices. Finally, for each vertex of the feedback vertex set of the partial solution, we need to identify one of its private cycles. To do so, we first guess the connected component of the potential final forest that "includes" such a private cycle, while additionally keeping track of the number of edges between the vertex and said component. ◀

4 ETH Lower Bound

In this section we present a lower bound on the complexity of solving Max Min FVS parameterized by vertex cover. Starting from a 3-SAT instance on n variables, we produce an equivalent Max Min FVS instance on a graph of vertex cover $O(n/\log n)$, hence any algorithm solving the latter problem in time $\mathrm{vc}^{o(\mathrm{vc})}n^{O(1)}$ would refute the ETH. As already mentioned, vertex cover is a very restrictive structural parameter, and due to known relationships of vertex cover with more general parameters, such as treedepth and treewidth, analogous lower bounds follow for these parameters. We first state the main theorem.

▶ **Theorem 2.** *There is no $\mathrm{vc}^{o(\mathrm{vc})}n^{O(1)}$ time algorithm for* Max Min FVS*, where* vc *denotes the size of the minimum vertex cover of the input graph, unless the ETH fails.*

Before we present the details of our construction, let us give some high-level intuition. Our goal is to "compress" an n-variable instance of 3-SAT, into an Max Min FVS instance with vertex cover roughly $n/\log n$. To this end, we will construct $\log n$ choice gadgets, each of which is supposed to represent $n/\log n$ variables, while contributing only $n/\log^2 n$ to the vertex cover. Hence, each vertex of each such gadget must be capable of representing roughly $\log n$ variables.

Our choice gadget may be thought of as a variation of a bipartite graph with sets L, R, of size roughly $n/\log^2 n$ and $\sqrt{n}$ respectively. If one naively tries to encode information in such a gadget by selecting which vertices of $L \cup R$ belong in an optimal solution, this would only give 2 choices per vertex, which is not efficient enough. Instead, we engineer things in a way that all vertices of $L \cup R$ must belong in the forest in an optimal solution, and the interesting choice for a vertex ℓ of L is with which vertex r of R we will place ℓ in the same component. In this sense, a vertex ℓ of L has $|R|$ choices, which is sufficient to encode the assignment for $\Omega(\log n)$ variables. What remains, then, is to add machinery that enforces this basic setup, and then clause checking vertices which for each clause verify that the clause is satisfied by testing if an ℓ vertex that represents one of its literals is in the same component as an r vertex that represents a satisfying assignment for the clause.

4.1 Preliminary Tools

Before we present the construction that proves Theorem 2, we give a variant of 3-SAT from which it will be more convenient to start our reduction, as well as a basic force gadget that we will use in our construction to ensure that some vertices must be placed in the forest in order to achieve an optimal solution.

3P3SAT. We first define a constrained version of 3-SAT, called 3-Partitioned-3-SAT (3P3SAT for short), and establish its hardness under the ETH.

> 3-Partitioned-3-SAT
> **Input:** A formula ϕ in 3-CNF form, together with a partition of the set of its variables V into three disjoint sets V_1, V_2, V_3, with $|V_i| = n$, such that no clause contains more than one variable from each V_i.
> **Task:** Determine whether ϕ is satisfiable.

▶ **Theorem 3.** ($\star$) *3-Partitioned-3-SAT cannot be decided in time $2^{o(n)}$, unless the ETH fails.*

Force gadgets. We now present a gadget that will ensure that a vertex u must be placed in the forest in any solution that finds a large minimal feedback vertex set. In the remainder, suppose that A is a sufficiently large value (we give a concrete value to A in the next section). When we say that we attach a *force gadget* to a vertex u, we introduce $A + 1$ new vertices $\bar{u}, u_1, \dots, u_A$ to the graph such that the vertices u_i form an independent set, while there exist edges $\{u, u_i\}, \{\bar{u}, u_i\}$ for all $i \in [A]$, as well as the edge $\{u, \bar{u}\}$. We refer to vertex $\bar{u}$ as the *gadget twin* of u, while the rest of the vertices will be referred to as the *gadget leaves* of u. Intuitively, the idea here is that if u (or $\bar{u}$) is contained in a minimal feedback vertex set, then none of the A leaves of the gadget can be taken, because these vertices cannot have private cycles. Hence, setting A to be sufficiently large will allow us to force u to be in the forest.

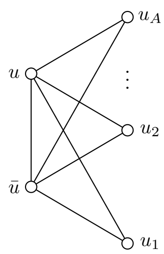


Figure 1 Force gadget attached to vertex u.

4.2 Construction

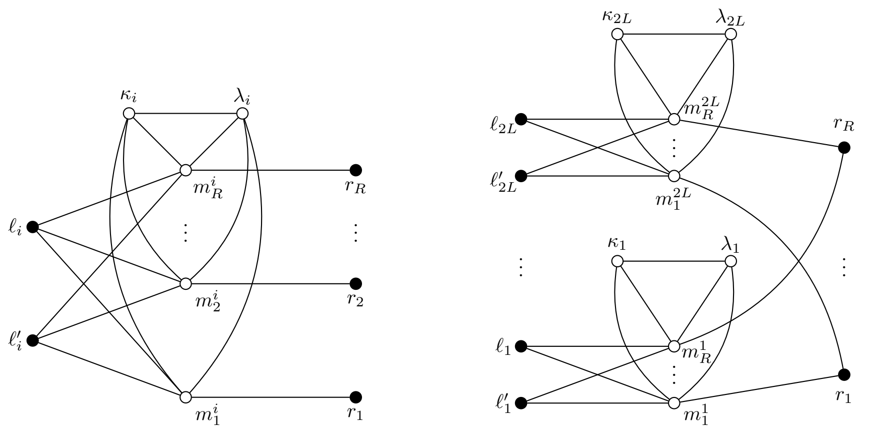
Let ϕ be a 3P3SAT instance of m clauses, where $|V_p| = n$ for $p \in [3]$ and, without loss of generality, assume that n is a power of 4 (this can be achieved by adding dummy variables to the instance if needed). Partition each variable set V_p to $\log n$ subsets V_p^q of size at most $\lceil \frac{n}{\log n} \rceil$, where $p \in [3]$ and $q \in [\log n]$. Let $L = \lceil \frac{n}{\log^2 n} \rceil$. Moreover, partition each variable subset V_p^q into $2L$ subsets $\mathcal{V}_\alpha^{p,q}$ of size as equal as possible, where $\alpha \in [2L]$. In the following we will omit p and q and instead use the notation $\mathcal{V}_\alpha$, whenever p, q are clear from the context. Define $R = \sqrt{n}$, $A = n^2 + m$ and $k = (4AL + AR + 2LR) \cdot 3 \log n + m$. We will proceed with the construction of a graph G such that G has a minimal feedback vertex set of size at least k if and only if ϕ is satisfiable.

For each variable subset V_p^q, we define the choice gadget graph G_p^q as follows:

- $V(G_p^q) = \{\ell_i, \ell'_i, \kappa_i, \lambda_i \mid i \in [2L]\} \cup \{r_j \mid j \in [R]\} \cup \{m_j^i \mid i \in [2L], j \in [R]\}$,
- all the vertices ℓ_i, ℓ'_i and r_j have an attached force gadget,
- for $i \in [2L]$, $N(\kappa_i) = M_i \cup \{\lambda_i\}$ and $N(\lambda_i) = M_i \cup \{\kappa_i\}$, where $M_i = \{m_j^i \mid j \in [R]\}$,
- for $i \in [2L]$ and $j \in [R]$, m_j^i has an edge with ℓ_i, ℓ'_i and r_j.

We will refer to the set $X_i = M_i \cup \{\kappa_i, \lambda_i\}$ as the *choice set* i.

Intuitively, one can think of this gadget as having been constructed as follows: we start with a complete bipartite graph that has on one side the vertices ℓ_i and on the other the vertices r_j; we subdivide each edge of this graph, giving the vertices m_j^i; for each $i \in [2L]$ we add $\ell'_i, \kappa_i, \lambda_i$, connect them to the same m_j^i vertices that ℓ_i is connected to and connect κ_i to λ_i; we attach force gadgets to all ℓ_i, ℓ'_i, r_j. Hence, as sketched before, the idea of this gadget is that the choice of a vertex ℓ_i is to pick an r_j with which it will be in the same component in the forest, and this will be expressed by picking one m_j^i that will be placed in the forest.



(a) Part of the construction concerning X_i. **(b)** The whole choice gadget graph G_p^q.

Figure 2 Black vertices have a force gadget attached.

Each vertex ℓ_α of G_p^q is used to represent a variable subset $\mathcal{V}_\alpha^{p,q} \subseteq V_p^q$ containing at most

$$|\mathcal{V}_\alpha^{p,q}| \leq \left\lceil \frac{|V^{p,q}|}{2L} \right\rceil \leq \left\lceil \frac{\left\lceil \frac{n}{\log n} \right\rceil}{2L} \right\rceil = \left\lceil \frac{n}{2L \log n} \right\rceil \leq \left\lceil \frac{n}{2\frac{n}{\log^2 n} \log n} \right\rceil = \left\lceil \frac{\log n}{2} \right\rceil = \frac{\log n}{2}$$

variables of ϕ, where we used Theorem 3.10 of [25], for $f(x) = x/2L$. We fix an arbitrary one-to-one mapping so that every vertex m_β^α, where $\beta \in [R]$, corresponds to a different assignment for this subset, which is dictated by which element of M_α was not included in the final feedback vertex set. Since $R = 2^{\log n/2} = \sqrt{n}$, the size of M_α is sufficient to uniquely encode all the different assignments of $\mathcal{V}_\alpha$.

Finally, introduce vertices c_i, where $i \in [m]$, each of which corresponds to a clause of ϕ, and define graph G as the union of these vertices as well as all graphs G_p^q, where $p \in [3]$ and $q \in [\log n]$. For a clause vertex c, add an edge to ℓ_α when $\mathcal{V}_\alpha$ contains a variable appearing in c, as well as to the vertices r_β for each such ℓ_α, such that $m_\beta^\alpha \notin S$ corresponds to an assignment of $\mathcal{V}_\alpha$ satisfying c, where S denotes a minimal feedback vertex set. Notice that since no clause contains multiple variables from the same variable set V_i, due to the refinement of the partition of the variables, it holds that all the variables of a clause will be represented by vertices appearing in distinct G_p^q.

4.3 Correctness

Having constructed the previously described instance (G, k) of Max Min FVS, it remains to prove its equivalence with the initial 3-Partitioned-3-SAT instance.

▶ **Lemma 4.** ($\star$) *Any minimal feedback vertex set S of G of size at least k has the following properties:*

(i) *S does not contain any vertex attached with a force gadget or its gadget twin,*
(ii) *$|M_i \setminus S| \leq 1$, for every G_p^q and $i \in [2L]$,*
(iii) *$|S \cap V(G_p^q)| = 4AL + AR + 2LR$,*
where $p \in [3]$ and $q \in [\log n]$.

▶ **Lemma 5.** ($\star$) *If ϕ has a satisfying assignment, then G has a minimal feedback vertex set of size at least k.*

▶ **Lemma 6.** ($\star$) *If G has a minimal feedback vertex set of size at least k, then ϕ has a satisfying assignment.*

▶ **Lemma 7.** ($\star$) *It holds that* $\mathrm{vc}(G) = O(n/\log n)$.

Using the previous lemmas, we can prove Theorem 2.

Proof of Theorem 2. Let ϕ be a 3-Partitioned-3-SAT formula. In polynomial time, we can construct a graph G such that, due to Lemmas 5 and 6, deciding if G has a minimal feedback vertex set of size at least k is equivalent to deciding if ϕ has a satisfying assignment. In that case, assuming there exists a $\mathrm{vc}^{o(\mathrm{vc})}$ algorithm for Max Min FVS, one could decide 3-Partitioned-3-SAT in time

$$\mathrm{vc}^{o(\mathrm{vc})} = \left(\frac{n}{\log n}\right)^{o(n/\log n)} = 2^{(\log n - \log\log n)o(n/\log n)} = 2^{o(n)},$$

which contradicts the ETH due to Theorem 3. ◀

Since for any graph G it holds that $\mathrm{tw}(G) \leq \mathrm{vc}(G)$, the following corollary holds.

▶ **Corollary 8.** *There is no $\mathrm{tw}^{o(\mathrm{tw})} n^{O(1)}$ time algorithm for Max Min FVS, where tw denotes the treewidth of the input graph, unless the ETH fails.*

5 Natural Parameter Algorithm

In this section we will present an FPT algorithm for Max Min FVS parameterized by the natural parameter, i.e. the size of the maximum minimal feedback vertex set k. The main theorem of this section is the following.

▶ **Theorem 9.** *Max Min FVS can be solved in time $9.34^k n^{O(1)}$.*

Structure of the Section. In Section 5.1 we define the closely related Annotated MMFVS problem, and prove that it remains NP-hard, even on some instances of specific form, called *path restricted instances*. Subsequently, we present an algorithm dealing with this kind of instances, which either returns a minimal feedback vertex set of size at least k or concludes that this is a No instance of Annotated MMFVS. Afterwards, in Section 5.2, we solve Max Min FVS by producing a number of instances of Annotated MMFVS and utilizing the previous algorithm, therefore proving Theorem 9.

Oversight of [23]. The algorithm of [23] performs a branching procedure which marks vertices as either belonging in the feedback vertex set or the remaining forest. The flaw is that the algorithm ceases the branching once k vertices have been identified as vertices of the feedback vertex set. However, this is not correct, since deciding if a given set S_0 can be extended into a minimal feedback vertex set $S \supseteq S_0$ is NP-complete and even W[1]-hard parameterized by $|S_0|$ [13]. Hence, identifying k vertices of the solution is not, in general, sufficient to produce a feasible solution and the algorithm of [23] is incomplete, because it does not explain how the guessed part of the feedback vertex set can be extended into a feasible minimal solution. Intuitively, the pitfall here is that, unlike other standard maximization problems, such as Max Clique, Max Min FVS is not monotone, that is, a graph that contains a feasible solution of size k is not guaranteed to contain a feasible solution of size $k-1$ (consider, for instance, a $K_{2,n}$).

5.1 Annotated MMFVS and Path Restricted Instances

First, we define the following closely related problem, denoted by Annotated MMFVS for short.

Annotated Maximum Minimal Feedback Vertex Set
Input: A graph $G = (V, E)$, disjoint sets $S, F \subseteq V$ where $S \cup F$ is a feedback vertex set of G, as well as an integer k.
Task: Determine whether there exists a minimal feedback vertex set S' of G of size $|S'| \geq k$ such that $S' \supseteq S$ and $S' \cap F = \emptyset$.

Remarks. Notice that if F is not a forest, then the corresponding instance always has a negative answer. For the rest of this section, let $U = V(G) \setminus (S \cup F)$. Moreover, let $H = \{s \in S \mid \deg_F(s) \geq 2 \text{ and } \deg_U(s) \leq 1\}$ denote the set of *good vertices* of S. An *interesting path* of $G[U]$ is a connected component of $G[U]$ such that for every vertex u belonging to said component, it holds that $\deg_{F \cup U}(u) = 2$. If every connected component of $G[U]$ is an interesting path, then this is a *path restricted instance*. Furthermore, given an instance $\mathcal{I}$, let $\text{ammfvs}(\mathcal{I})$ be equal to 1 if it is a Yes instance and 0 otherwise.

Let $\mathcal{I} = (G, S, F, k)$ be a path restricted instance of Annotated MMFVS. We will present an algorithm that either returns a minimal feedback vertex set $S' \subseteq S \cup U$ of G of size at least k or concludes that this is a No instance of Annotated MMFVS. Notice that Annotated MMFVS remains NP-hard even on such instances, as dictated by Theorem 10. Therefore, we should not expect to solve path restricted instances of Annotated MMFVS in polynomial time.

▶ **Theorem 10.** ($\star$) *Annotated MMFVS is NP-hard on path restricted instances, even if all the paths are of length* 2.

We proceed by presenting the main algorithm of this subsection, which will be essential in proving Theorem 9.

▶ **Theorem 11.** (⋆) *Let $\mathcal{I} = (G, S, F, k)$ be a path restricted instance of* Annotated *MMFVS, and let g denote the number of its good vertices. There is an algorithm running in time $O(3^{k-g} n^{O(1)})$ which either returns a minimal feedback vertex set $S' \subseteq S \cup U$ of G of size at least k or concludes that $\mathcal{I}$ is a No instance of* Annotated *MMFVS.*

Proof sketch. The main idea of the algorithm lies on the fact that we can efficiently handle instances where either $k = 0$ or $S = \emptyset$. Towards this, we will employ a branching strategy that, as long as S remains non empty, new instances with reduced k are produced. Prior to performing branching, we first observe that we can efficiently deal with the good vertices. Afterwards, by employing said branching strategy, in every step we decide which vertex will be counted towards the k required, thereby reducing parameter k on each iteration. If at some point $k = 0$ or $S = \emptyset$, it remains to decide whether this comprises a viable solution S'. Notice that S' may not be a solution for the annotated instance, since even if $|S'| \geq k$, it does not necessarily hold that $S' \supseteq S$. ◀

5.2 Algorithm for Max Min FVS

We start by presenting a high level sketch of the algorithm for Max Min FVS. The starting point is a minimal feedback vertex set S_0 of G. Note that such a set can be obtained in polynomial time, while if it is of size at least k, we are done. Therefore, assume that $|S_0| < k$. Then, assuming there exists a minimal feedback vertex set S^*, where $|S^*| \geq k$ and $F^* = V(G) \setminus S^*$, we will guess $S_0 \cap S^*$, thereby producing instances $\mathcal{I}_0 = (G, S_0 \cap S^*, S_0 \cap F^*, k)$ of Annotated MMFVS. Subsequently, we will establish a number of *safe* reduction rules, which do not affect the answer of the instances. We will present a measure of progress μ, which guarantees that if an instance $\mathcal{I} = (G, S, F, k)$ of Annotated MMFVS has $\mu(\mathcal{I}) \leq 1$, then G has a minimal feedback vertex set $S' \subseteq S \cup U$ of size at least k. Then, we will employ a branching strategy which, given $\mathcal{I}_i$, will produce instances $\mathcal{I}^1_{i+1}, \mathcal{I}^2_{i+1}$ of lesser measure of progress, such that $\mathcal{I}_i$ is a Yes instance if and only if at least one of $\mathcal{I}^1_{i+1}, \mathcal{I}^2_{i+1}$ is also a Yes instance. If we can no further apply our branching strategy, and the measure of progress remains greater than 1, then it holds that $\mathcal{I}$ is a path restricted instance and Theorem 11 applies.

Measure of progress. Let $\mathcal{I} = (G, S, F, k)$ be an instance of Annotated MMFVS. We define as $\mu(\mathcal{I}) = k + cc(F) - g - p$ its *measure of progress*, where

- $cc(F)$ denotes the number of connected components of F,
- g denotes the number of good vertices of S,
- p denotes the number of interesting paths of $G[U]$.

It holds that if $\mu(\mathcal{I}) \leq 1$, then the underlying Max Min FVS instance has a positive answer, which does not necessarily respect the constraints dictated by the annotated version.

▶ **Lemma 12.** (⋆) *Let $\mathcal{I} = (G, S, F, k)$ be an instance of* Annotated *MMFVS, where $\mu(\mathcal{I}) \leq 1$. Then, G has a minimal feedback vertex set $S' \subseteq S \cup U$ of size at least k.*

Reduction rules. In the following, we will describe some reduction rules which do not affect the answer of an instance of Annotated MMFVS, while not increasing its measure of progress.

▶ **Lemma 13.** ($\star$) *Let $G = (V, E)$ be a (multi)graph and $uv \in E(G)$. Then, G is acyclic if and only if G/uv is acyclic.*

Rule 1. Let $\mathcal{I} = (G, S, F, k)$ be an instance of Annotated MMFVS, $u, v \in F$ and $uv \in E$. Then, replace $\mathcal{I}$ with $\mathcal{I}' = (G', S, F', k)$, where $G' = G/uv$ occurs from the contraction of u and v into w, while $F' = (F \cup \{w\}) \setminus \{u, v\}$.

Rule 2. Let $\mathcal{I} = (G, S, F, k)$ be an instance of Annotated MMFVS, $u \in U$ and $\deg_{F\cup U}(u) = 0$. Then, replace $\mathcal{I}$ with $\mathcal{I}' = (G - u, S, F, k)$.

Rule 3. Let $\mathcal{I} = (G, S, F, k)$ be an instance of Annotated MMFVS, $u \in U$ and $\deg_{F\cup U}(u) = 1$, while $v \in N(u) \cap (F \cup U)$. Then, replace $\mathcal{I}$ with $\mathcal{I}' = (G', S, F', k)$, where $G' = G/uv$ occurs from the contraction of u and v into w, while $F' = (F \cup \{w\}) \setminus \{v\}$ if $v \in F$, and $F' = F$ otherwise.

▶ **Lemma 14.** ($\star$) *Applying rules 1, 2 and 3 does not change the outcome of the algorithm and does not increase the measure of progress.*

After exhaustively applying the aforementioned rules, it holds that $\forall u \in U$, $\deg_{F\cup U}(u) \geq 2$, i.e. $G[U]$ is a forest containing trees, all the leaves of which have at least one edge to F. Moreover, $G[F]$ comprises an independent set. We proceed with a branching strategy that produces instances of Annotated MMFVS of reduced measure of progress. If at some point $\mu \leq 1$, then Lemma 12 can be applied.

Branching strategy. Let $\mathcal{I} = (G, S, F, k)$ be an instance of Annotated MMFVS, on which all of the reduction rules have been applied exhaustively, thus, it holds that a) $\forall u \in U$, $\deg_{F\cup U}(u) \geq 2$ and b) F is an independent set.

Define $u \in U$ to be an *interesting* vertex if $\deg_{F\cup U}(u) \geq 3$. As already noted, $G[U]$ is a forest, the leaves of which all have an edge towards F, otherwise Rule 3 could still be applied. Consider a root for each tree of $G[U]$. For some tree T, let v be an interesting vertex at maximum distance from the corresponding root, i.e. v is an interesting vertex of maximum height. Notice that such a tree cannot be an interesting path. We branch depending on whether u is in the feedback vertex set or not. Towards this end, let $S' = S \cup \{v\}$ and $F' = F \cup \{v\}$, while $\mathcal{I}_1 = (G, S', F, k)$ and $\mathcal{I}_2 = (G, S, F', k)$. It holds that $\mathcal{I}$ is a Yes instance if and only if at least one of $\mathcal{I}_1, \mathcal{I}_2$ is a Yes instance, while if $G[F']$ contains a cycle, $\mathcal{I}_2$ is a No instance and we discard it. We replace $\mathcal{I}$ with the instances $\mathcal{I}_1$ and $\mathcal{I}_2$.

▶ **Lemma 15.** ($\star$)*The branching strategy produces instances of reduced measure of progress, without reducing the number of good vertices. Additionally, whenever the branching places a vertex on the feedback vertex set, this vertex is good.*

Complexity. Starting from an instance (G, k) of Max Min FVS, we produce a minimal feedback vertex set S_0 of G in polynomial time. If $|S_0| \geq k$, we are done. Alternatively, we produce instances of Annotated MMFVS by guessing the intersection of S_0 with some minimal feedback vertex set of G of size at least k. Let $\mathcal{I} = (G, S, F, k)$ be one such instance. It holds that $\mu(\mathcal{I}) \leq k + c$, where $c = cc(F)$, therefore the branching will perform at most $k + c$ steps. Notice that, at any step of the branching procedure, the number of good vertices never decreases. Now, consider a path restricted instance $\mathcal{I}' = (G', S', F', k)$ resulting from branching starting on $\mathcal{I}$, on which branching, exactly ℓ times a vertex was placed in the feedback vertex set, therefore $|S'| - |S| = \ell$. There are at most $\binom{k+c}{\ell}$ different such instances, each of which has at least ℓ good vertices, thus Theorem 11 requires time at most $3^{k-\ell}n^{O(1)}$. Since $0 \le \ell \le k+c$, and there are at most $\binom{k}{c}$ different instances $\mathcal{I}$, the algorithm runs in time $9.34^k n^{O(1)}$, since

$$\sum_{c=0}^{k}\binom{k}{c}\sum_{\ell=0}^{k+c}\binom{k+c}{\ell}3^{k-\ell} = 3^k\sum_{c=0}^{k}\binom{k}{c}\sum_{\ell=0}^{k+c}\binom{k+c}{\ell}3^{-\ell} = 3^k\sum_{c=0}^{k}\binom{k}{c}\left(\frac{4}{3}\right)^{k+c}$$
$$= 4^k\sum_{c=0}^{k}\binom{k}{c}\left(\frac{4}{3}\right)^{c} = 4^k\left(\frac{7}{3}\right)^{k} \le 9.34^k.$$

6 The Extension Problem

In this section we consider the following extension problem:

> MINIMAL FVS EXTENSION
> **Input:** A graph $G = (V, E)$ and a set $S \subseteq V$.
> **Task:** Determine whether there exists $S^* \supseteq S$ such that S^* is a minimal feedback vertex set of G.

Observe that this is a special case of ANNOTATED MMFVS, since we essentially set $F = \emptyset$ and do not care about the size of the produced solution, albeit with the difference that now we will not focus on the case where $V \setminus S$ is already acyclic. This extension problem was already shown to be W[1]-hard parameterized by $|S|$ by Casel et al. [13]. One question that was left open, however, was whether it is solvable in polynomial time for fixed $|S|$, that is, whether it belongs in the class XP. Superficially, this seems somewhat surprising, because for the closely related MAXIMUM MINIMAL VERTEX COVER and UPPER DOMINATING SET problems, membership of the extension problem in XP is almost trivial: it suffices to guess for each $v \in S$ a private edge or vertex that is only dominated by v, remove from consideration other vertices that dominate this private edge or vertex, and then attempt to find any feasible solution. The reason that this strategy does not seem to work for feedback vertex set is that for each $v \in S$ we would have to guess a private cycle. Since a priori we have no bound on the length of such a cycle, there is no obvious way to achieve this task in $n^{f(k)}$ time.

Though we do not settle the complexity of the extension problem for fixed k, we provide evidence that obtaining a polynomial time algorithm would be a challenging task, because it would imply a similar algorithm for the k-IN-A-TREE problem. In the latter, we are given a graph G and a set T of k terminals and are asked to find a set T^* such that $T \subseteq T^*$ and $G[T^*]$ is a tree [15, 29].

▶ **Theorem 16.** *k-IN-A-TREE parameterized by k is fpt-reducible to MINIMAL FVS EXTENSION parameterized by the size of the given set.*

Proof. Consider an instance $G = (V, E)$ of k-IN-A-TREE, with terminal set T. Let $T = \{t_1, \dots, t_k\}$. We add to the graph $k-1$ new vertices, $s_1, \dots, s_{k-1}$ and connect each s_i to t_i and to t_{i+1}, for $i \in [k-1]$. We set $S = \{s_1, \dots, s_{k-1}\}$. This completes the construction. Clearly, this reduction preserves the value of the parameter.

To see correctness, suppose first that a tree $T^* \supseteq T$ exists in G. We set $S_1 = S \cup (V \setminus T^*)$ in the new graph. S_1 is a feedback vertex set, because removing it from the graph leaves T^*, which is a tree. S_1 contains S. Furthermore, if S_1 is not minimal, we greedily remove from it arbitrary vertices until we obtain a minimal feedback vertex set S_2. We claim that S_2 must still contain S. Indeed, each vertex s_i, for $i \in [k-1]$ has a private cycle, since its neighbors $t_i, t_{i+1} \in T^*$. For the converse direction, if there exists in the new graph a minimal feedback vertex set S^* that contains S, then the remaining forest $F^* = V \setminus S^*$ must contain T, since each vertex of S must have a private cycle in the forest, and vertices of S have degree 2. Furthermore, all vertices of T must be in the same component of F^*, because to obtain a private cycle for s_i, we must have a path from t_i to t_{i+1} in F^*, for all $i \in [k-1]$. Therefore, in this case we have found an induced tree in G that contains all terminals. ◀

7 Conclusions and Open Problems

We have precisely determined the complexity of Max Min FVS with respect to structural parameters from vertex cover to treewidth as being slightly super-exponential. One natural question to consider would then be to examine if the same complexity can be achieved when the problem is parameterized by clique-width. Regarding the complexity of the extension problem for sets of fixed size k, we have shown that this is at least as hard as the well-known (and wide open) k-in-a-Tree problem. Barring a full resolution of this question, it would also be interesting to ask if the converse reduction also holds, which would prove that the two problems are actually equivalent.

References

1 Hassan AbouEisha, Shahid Hussain, Vadim V. Lozin, Jérôme Monnot, Bernard Ries, and Viktor Zamaraev. Upper domination: Towards a dichotomy through boundary properties. *Algorithmica*, 80(10):2799–2817, 2018. doi:10.1007/s00453-017-0346-9.

2 Júlio Araújo, Marin Bougeret, Victor A. Campos, and Ignasi Sau. Introducing lop-kernels: A framework for kernelization lower bounds. *Algorithmica*, 84(11):3365–3406, 2022. doi:10.1007/s00453-022-00979-z.

3 Júlio Araújo, Marin Bougeret, Victor A. Campos, and Ignasi Sau. Parameterized complexity of computing maximum minimal blocking and hitting sets. *Algorithmica*, 85(2):444–491, 2023. doi:10.1007/s00453-022-01036-5.

4 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In *SODA*, pages 951–970. SIAM, 2020. doi:10.1137/1.9781611975994.57.

5 Cristina Bazgan, Ljiljana Brankovic, Katrin Casel, Henning Fernau, Klaus Jansen, Kim-Manuel Klein, Michael Lampis, Mathieu Liedloff, Jérôme Monnot, and Vangelis Th. Paschos. The many facets of upper domination. *Theor. Comput. Sci.*, 717:2–25, 2018. doi:10.1016/j.tcs.2017.05.042.

6 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Yota Otachi. Grundy distinguishes treewidth from pathwidth. *SIAM Journal on Discrete Mathematics*, 36(3):1761–1787, 2022. doi:10.1137/20M1385779.

7 Benjamin Bergougnoux, Édouard Bonnet, Nick Brettell, and O-joung Kwon. Close relatives of feedback vertex set without single-exponential algorithms parameterized by treewidth. In *IPEC*, volume 180 of *LIPIcs*, pages 3:1–3:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.IPEC.2020.3.

8 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.

9 Marthe Bonamy, Lukasz Kowalik, Jesper Nederlof, Michal Pilipczuk, Arkadiusz Socala, and Marcin Wrochna. On directed feedback vertex set parameterized by treewidth. In *WG*, volume 11159 of *Lecture Notes in Computer Science*, pages 65–78. Springer, 2018. doi:10.1007/978-3-030-00256-5_6.

10 Édouard Bonnet, Nick Brettell, O-joung Kwon, and Dániel Marx. Generalized feedback vertex set problems on bounded-treewidth graphs: Chordality is the key to single-exponential parameterized algorithms. *Algorithmica*, 81(10):3890–3935, 2019. doi:10.1007/s00453-019-00579-4.

11 Édouard Bonnet, Michael Lampis, and Vangelis Th. Paschos. Time-approximation trade-offs for inapproximable problems. *J. Comput. Syst. Sci.*, 92:171–180, 2018. doi:10.1016/j.jcss.2017.09.009.

12 Nicolas Boria, Federico Della Croce, and Vangelis Th. Paschos. On the max min vertex cover problem. *Discret. Appl. Math.*, 196:62–71, 2015. doi:10.1016/j.dam.2014.06.001.

13 Katrin Casel, Henning Fernau, Mehdi Khosravian Ghadikolaei, Jérôme Monnot, and Florian Sikora. On the complexity of solution extension of optimization problems. *Theor. Comput. Sci.*, 904:48–65, 2022. doi:10.1016/j.tcs.2021.10.017.

14 Juhi Chaudhary, Sounaka Mishra, and B. S. Panda. Minimum maximal acyclic matching in proper interval graphs. In *CALDAM*, volume 13947 of *Lecture Notes in Computer Science*, pages 377–388. Springer, 2023. doi:10.1007/978-3-031-25211-2_29.

15 Maria Chudnovsky and Paul D. Seymour. The three-in-a-tree problem. *Comb.*, 30(4):387–417, 2010. doi:10.1007/s00493-010-2334-4.

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

17 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Trans. Algorithms*, 18(2):17:1–17:31, 2022. doi:10.1145/3506707.

18 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate texts in mathematics.* Springer, 2017. doi:10.1007/978-3-662-53622-3.

19 Gabriel L. Duarte, Hiroshi Eto, Tesshu Hanaka, Yasuaki Kobayashi, Yusuke Kobayashi, Daniel Lokshtanov, Lehilton L. C. Pedrosa, Rafael C. S. Schouery, and Uéverton S. Souza. Computing the largest bond and the maximum connected cut of a graph. *Algorithmica*, 83(5):1421–1458, 2021. doi:10.1007/s00453-020-00789-1.

20 Louis Dublois, Tesshu Hanaka, Mehdi Khosravian Ghadikolaei, Michael Lampis, and Nikolaos Melissinos. (in)approximability of maximum minimal FVS. *J. Comput. Syst. Sci.*, 124:26–40, 2022. doi:10.1016/j.jcss.2021.09.001.

21 Louis Dublois, Michael Lampis, and Vangelis Th. Paschos. Upper dominating set: Tight algorithms for pathwidth and sub-exponential approximation. *Theor. Comput. Sci.*, 923:271–291, 2022. doi:10.1016/j.tcs.2022.05.013.

22 Fabio Furini, Ivana Ljubic, and Markus Sinnl. An effective dynamic programming algorithm for the minimum-cost maximal knapsack packing problem. *Eur. J. Oper. Res.*, 262(2):438–448, 2017. doi:10.1016/j.ejor.2017.03.061.

23 Ajinkya Gaikwad, Hitendra Kumar, Soumen Maity, Saket Saurabh, and Shuvam Kant Tripathi. Maximum minimal feedback vertex set: A parameterized perspective. *CoRR*, abs/2208.01953, 2022. doi:10.48550/arXiv.2208.01953.

24 Laurent Gourvès, Jérôme Monnot, and Aris Pagourtzis. The lazy bureaucrat problem with common arrivals and deadlines: Approximation and mechanism design. In *FCT*, volume 8070 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 2013. doi:10.1007/978-3-642-40164-0_18.

25 Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science, 2nd Ed.* Addison-Wesley, 1994.

26 Tesshu Hanaka, Yasuaki Kobayashi, Yusuke Kobayashi, and Tsuyoshi Yagita. Finding a maximum minimal separator: Graph classes and fixed-parameter tractability. *Theor. Comput. Sci.*, 865:131–140, 2021. doi:10.1016/j.tcs.2021.03.006.

27 Ararat Harutyunyan, Michael Lampis, and Nikolaos Melissinos. Digraph coloring and distance to acyclicity. In *STACS*, volume 187 of *LIPIcs*, pages 41:1–41:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.41.

28 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

29 Kai-Yuan Lai, Hsueh-I Lu, and Mikkel Thorup. Three-in-a-tree in near linear time. In *STOC*, pages 1279–1292. ACM, 2020. doi:10.1145/3357713.3384235.

30 Michael Lampis. Minimum stable cut and treewidth. In *ICALP*, volume 198 of *LIPIcs*, pages 92:1–92:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.92.

31 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. *SIAM J. Comput.*, 47(3):675–702, 2018. doi:10.1137/16M1104834.

32 Sounaka Mishra and Kripasindhu Sikdar. On the hardness of approximating some np-optimization problems related to minimum linear ordering problem. *RAIRO Theor. Informatics Appl.*, 35(3):287–309, 2001. doi:10.1051/ita:2001121.

33 Michal Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In *MFCS*, volume 6907 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 2011. doi:10.1007/978-3-642-22993-0_47.

34 Meirav Zehavi. Maximum minimal vertex cover parameterized by vertex cover. *SIAM J. Discret. Math.*, 31(4):2440–2456, 2017. doi:10.1137/16M109017X.