(In)approximability of Maximum Minimal FVS[☆]

Louis Dublois[a], Tesshu Hanaka[b], Mehdi Khosravian Ghadikolaei[a], Michael Lampis[a], Nikolaos Melissinos[a]

[a] *Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243, LAMSADE, Paris, France*
[b] *Nagoya University, Nagoya, Japan*

Abstract

We study the approximability of the NP-complete Maximum Minimal Feedback Vertex Set problem. Informally, this natural problem seems to lie in an intermediate space between two more well-studied problems of this type: Maximum Minimal Vertex Cover, for which the best achievable approximation ratio is $\sqrt{n}$, and Upper Dominating Set, which does not admit any $n^{1-\epsilon}$ approximation. We confirm and quantify this intuition by showing the first non-trivial polynomial time approximation for Max Min FVS with a ratio of $O(n^{2/3})$, as well as a matching hardness of approximation bound of $n^{2/3-\epsilon}$, improving the previous known hardness of $n^{1/2-\epsilon}$. The approximation algorithm also gives a cubic kernel when parameterized by the solution size. Along the way, we also obtain an $O(\Delta)$-approximation and show that this is asymptotically best possible, and we improve the bound for which the problem is NP-hard from $\Delta \geq 9$ to $\Delta \geq 6$.

Having settled the problem's approximability in polynomial time, we move to the context of super-polynomial time. We devise a generalization of our approximation algorithm which, for any desired approximation ratio r, produces an r-approximate solution in time $n^{O(n/r^{3/2})}$. This time-approximation trade-off is essentially tight: we show that under the ETH, for

[☆] A preliminary version of the paper appeared in Proceedings of the 31st International Symposium on Algorithms and Computation (ISAAC 2020), Vol. 181 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 3:1–3:14 (2020) [20]. This version contains all the proofs and some new observations about the kernelization of the problem.

Email addresses: `louis.dublois@gmail.com` (Louis Dublois), `hanaka@nagoya-u.jp` (Tesshu Hanaka), `m.khosravian@gmail.com` (Mehdi Khosravian Ghadikolaei), `michail.lampis@dauphine.fr` (Michael Lampis), `nikolaos.melissinos@dauphine.eu` (Nikolaos Melissinos)

any ratio r and $\epsilon > 0$, no algorithm can r-approximate this problem in time $n^{O((n/r^{3/2})^{1-\epsilon})}$, hence we precisely characterize the approximability of the problem for the whole spectrum between polynomial and sub-exponential time, up to an arbitrarily small constant in the second exponent.

Keywords: Approximation Algorithms, ETH, Inapproximability

---

1. Introduction

In a graph $G = (V, E)$, a set $S \subseteq V$ is called a *feedback vertex set* (fvs for short) if the subgraph induced by $V \setminus S$ is a forest. Typically, fvs is studied with a minimization objective: given a graph we are interested in finding the best (that is, smallest) fvs. In this paper we are interested in an objective which is, in a sense, the inverse: we seek an fvs S which is as *large* as possible, while still being minimal. We call this problem Max Min FVS.

MaxMin and MinMax versions of many famous optimization problems have recently attracted much interest in the literature (we give references below) and Max Min FVS can be seen as a member of this framework. Although the initial motivation for studying such problems was a desire to analyze the worst possible performance of a naive heuristic, these problems have gradually been revealed to possess a rich combinatorial structure that makes them interesting in their own right. Our goal in this paper is to show that Max Min FVS displays an interesting complexity behavior with respect to its approximability.

Our motivation for focusing on Max Min FVS is the contrast between two of its more well-studied cousins: the Max Min Vertex Cover (Max Min VC) and Upper Dominating Set (UDS) problems (we give references below), where the objective is to find the largest minimal vertex cover or dominating set, respectively. At first glance, one would expect Max Min VC to be the easier of these two problems: both problems can be seen as trying to find the largest minimal hitting set of a hypergraph, but in the case of Max Min VC the hypergraph has a very restricted structure, while in UDS the hypergraph is essentially arbitrary. This intuition turns out to be correct: while UDS admits no $n^{1-\epsilon}$-approximation [5], Max Min VC admits a $\sqrt{n}$-approximation (but no $n^{1/2-\epsilon}$-approximation) [9].

This background leads us to the natural question of the approximability of Max Min FVS. On an intuitive level, one may be tempted to think that this problem should be harder than Max Min VC, since hitting cycles is more complex than hitting edges, but easier than UDS, since hitting cycles still offers us more structure than an arbitrary hypergraph. However, to the best of our knowledge, no $n^{1-\epsilon}$-approximation algorithm is currently known for Max Min FVS (so the problem could be as hard as UDS), and the best hardness of approximation bound known is $n^{1/2-\epsilon}$ [38] (so the problem could be as easy as Max Min VC).

Our main contribution in this paper is to fully answer this question, confirming and precisely quantifying the intuition that Max Min FVS is a problem that lies “between” Max Min VC and UDS: we give a polynomial-time approximation algorithm with ratio $O(n^{2/3})$ and a hardness of approximation reduction which shows that (unless P = NP) no polynomial-time algorithm can obtain a ratio of $n^{2/3-\epsilon}$, for any $\epsilon > 0$. This completely settles the approximability of the problem in polynomial time. Along the way, we also prove that Max Min FVS admits a cubic kernel when parameterized by the solution size, give an approximation algorithm with ratio $O(\Delta)$, show that no algorithm can achieve ratio $\Delta^{1-\epsilon}$, for any $\epsilon > 0$, and improve the best known NP-completeness proof for Max Min FVS from $\Delta \geq 9$ [38] to $\Delta \geq 6$, where Δ is the maximum degree of the input graph.

One interesting aspect of our results is that they have an interpretation from extremal combinatorics which nicely mirrors the situation for Max Min VC. Recall that a corollary of the $\sqrt{n}$-approximation for Max Min VC [9] is that any graph without isolated vertices has a minimal vertex cover of size at least $\sqrt{n}$, and this is tight (see Remark 3). Hence, the algorithm only needs to trivially preprocess the graph (deleting isolated vertices) and then find this set, which is guaranteed to exist. Our algorithms can be seen in a similar light: we prove that if one applies two almost trivial pre-processing rules to a graph (deleting leaves and contracting edges between degree-two vertices), a minimal fvs of size at least $n^{1/3}$ (and $\Omega(n/\Delta)$) is always guaranteed to exist, and this is tight (Corollary 1 and Remark 2). Thus, the approximation ratio of $n^{2/3}$ is automatically guaranteed for any graph where we exhaustively apply these very simple rules and our algorithms only have to work to construct the promised set. This makes it somewhat remarkable that the ratio of $n^{2/3}$ turns out to be best possible.

Having settled the approximability of Max Min FVS in polynomial time, we consider the question of how much time needs to be invested if one wishes to guarantee an approximation ratio of r (which may depend on n) where $r < n^{2/3}$. This type of time-approximation trade-off was extensively studied by Bonnet et al. [8], who showed that Max Min VC admits an r-approximation in time $2^{O(n/r^2)}$ and this is optimal under the randomized ETH.

For Max Min FVS we cannot hope to obtain a trade-off with performance exponential in n/r^2, as this implies a polynomial-time $\sqrt{n}$-approximation. It therefore seems more natural to aim for a running time exponential in $n/r^{3/2}$. Indeed, generalizing our polynomial-time approximation algorithm, we show that we can achieve an r-approximation in time $n^{O(n/r^{3/2})}$. Although this algorithm reuses some ingredients from our polynomial-time approximation, it is significantly more involved, as it is no longer sufficient to compare the size of our solution to n. We complement our result with a lower bound showing that our algorithm is essentially best possible under the randomized ETH for any r (not just for polynomial time), or more precisely that the exponent of the running time of our algorithm can only be improved by $n^{o(1)}$ factors.

Related work To the best of our knowledge, Max Min FVS was first considered by Mishra and Sikdar [38], who showed that the problem does not admit an $n^{1/2-\epsilon}$ approximation (unless P = NP), and that it remains APX-hard for $\Delta \geq 9$. On the other hand, UDS and Max Min VC are well-studied problems, both in the context of approximation and in the context of parameterized complexity [1, 5, 9, 11, 13, 14, 19, 30, 35, 41, 43, 21]. Many other classical optimization problems have recently been studied in the MaxMin or MinMax framework, such as Max Min Separator [27], Max Min Cut [23], Min Max Knapsack (also known as the Lazy Bureaucrat Problem) [3, 25, 26], and some variants of Max Min Edge Cover [37, 28]. Some problems in this area also arise naturally in other forms and have been extensively studied, such as Min Max Matching (also known as Edge Dominating Set [34]), Grundy Coloring, which can be seen as a Max Min version of Coloring [2, 6], and Max Min VC in hypergraphs, which is known as Upper Transversal[39, 31, 32, 33].

The idea of designing super-polynomial time approximation algorithms which obtain guarantees better than those possible in polynomial time has attracted much attention in the last decade [4, 10, 16, 18, 22, 24, 36]. As mentioned, the result closest to the time-approximation trade-off we give in this paper is the approximation algorithm for Max Min VC given by Bonnet et al. [8]. It is important to note that such trade-offs are only generally known to be tight up to poly-logarithmic factors in the exponent of the running time. As explained in [8], current lower bound techniques can rule out improvements in the running time that shave at least n^ϵ from the exponent, but not improvements which shave poly-logarithmic factors, due to the state of the art in quasi-linear PCP constructions. Indeed, such improvements are sometimes possible [4] and are conceivable for Max Min VC and Max Min FVS. Such lower bounds rely on the (randomized) Exponential Time Hypothesis (ETH), which states that there is no (randomized) algorithm for 3-SAT running in time $2^{o(n)}$.

2. Preliminaries

We use standard graph-theoretic notation and only consider simple (without parallel edges) loop-less graphs. For a graph $G = (V, E)$ and $S \subseteq V$ we denote by $G[S]$ the graph induced by S. For $u \in V$, $G - u$ is the graph $G[V \setminus \{u\}]$. We write $N(u)$ to denote the set of neighbors of u and $d(u) = |N(u)|$ to denote its degree. For $S \subseteq V$, $N(S) = \bigcup_{u \in S} N(u) \setminus S$. We use $\Delta(G)$ (or simply Δ) to denote the maximum degree of G. For $uv \in E$ the graph G/uv is the graph obtained by contracting the edge uv, that is, replacing u, v by a new vertex connected to $N(\{u, v\})$. In this paper we will only apply this operation when $N(u) \cap N(v) = \emptyset$, so the result will always be a simple graph.

A forest is a graph that does not contain cycles. A feedback vertex set (fvs for short) is a set $S \subseteq V$ such that $G[V \setminus S]$ is a forest. An fvs S is minimal if no proper subset of S is an fvs. It is not hard to see that if S is minimal, then every $u \in S$ has a *private cycle*, that is, there exists a cycle in $G[(V \setminus S) \cup \{u\}]$, which goes through u. A vertex u of a feedback vertex set S that does not have a private cycle (that is, $S \setminus \{u\}$ is also an fvs), is called *redundant*. For a given fvs S, we call the set $F = V \setminus S$ the corresponding induced forest. If S is minimal, then F is maximal.

The main problem we are interested in is Max Min FVS: given a graph $G = (V, E)$, find a minimal fvs of G of maximum size. Since this problem is NP-hard, we will be interested in approximation algorithms. An approximation algorithm with ratio $r \geq 1$ (which may depend on n, the number of vertices of the graph) is an algorithm which, given a graph G, returns a solution of size at least $\frac{\text{mmfvs}(G)}{r}$, where $\text{mmfvs}(G)$ is the size of the largest minimal fvs of G.

We make two basic observations about our problem: deleting vertices or contracting edges can only decrease the size of the optimal solution.

Lemma 1. *Let $G = (V, E)$ be a graph and $u \in V$. Then,* $\text{mmfvs}(G) \geq \text{mmfvs}(G - u)$*. Furthermore, given any minimal feedback vertex set S of $G - u$, it is possible to construct in polynomial time a minimal feedback vertex set of G of the same or larger size.*

Proof. Let S be a minimal fvs of $G - u$. We observe that $S \cup \{u\}$ is an fvs of G. If $S \cup \{u\}$ is minimal, we are done. If not, we delete vertices from it until it becomes minimal. We now note that the only vertex which may be deleted in this process is u, since all vertices of S have a private cycle in $G - u$ (that is, a cycle not intersected by any other vertex of S). Hence, the resulting set is a superset of S. $\square$

Lemma 2. *Let $G = (V, E)$ be a graph, $u, v \in V$ with $N(u) \cap N(v) = \emptyset$ and $uv \in E$. Then* $\mathrm{mmfvs}(G) \geq \mathrm{mmfvs}(G/uv)$*. Furthermore, given any minimal feedback vertex set S of G/uv, it is possible to construct in polynomial time a minimal feedback vertex set of G of the same or larger size.*

Proof. Before we prove the Lemma we note that the contraction operation, under the condition that $N(u) \cap N(v) = \emptyset$, preserves acyclicity in a strong sense: G is acyclic if and only if G/uv is acyclic. Indeed, if we contract an edge that is part of a cycle, this cycle must have length at least 4 since $N(u) \cap N(v) = \emptyset$, and will therefore give a cycle in G/uv. Of course, contractions never create cycles in acyclic graphs.

Let $G' = G/uv$ and w be the vertex of G' which has replaced u, v. Let $V' = V(G')$, and S be a minimal fvs of G'. We have two cases: $w \in S$ or $w \notin S$.

In case $w \in S$, we start with the set $S' = (S \setminus \{w\}) \cup \{u, v\}$. It is not hard to see that S' is an fvs of G. Furthermore, no vertex of $S' \setminus \{u, v\}$ is redundant: for all $z \in S \setminus \{w\}$, there is a cycle in $G'[(V' \setminus S) \cup \{z\}]$, therefore there is also a cycle in $G[(V \setminus S') \cup \{z\}]$. Furthermore, we claim that $S' \setminus \{u, v\}$ is not a valid fvs. Indeed, there must be a cycle contained (due to minimality) in $G_1 = G'[(V' \setminus S) \cup \{w\}]$. Therefore, if there is no cycle in $G_2 = G[(V \setminus S') \cup \{u, v\}]$, we get a contradiction, as G_1 can be obtained from G_2 by contracting the edge uv and contracting edges preserves acyclicity. We conclude that even if S' is not minimal, if we remove vertices until it becomes minimal, we will remove at most one vertex, so the size of the fvs obtained is at least $|S|$.

In case $w \notin S$, we will return the same set S. Let $F = V \setminus S, F' = V' \setminus S$. By definition, $G'[F']$ is acyclic. To see that $G[F]$ is also a forest, we note that $G'[F']$ is obtained from $G[F]$ by contracting uv, and as we noted in the beginning, the contractions we use strongly preserve acyclicity. To see that S is minimal, take $z \in S$ and consider the graphs $G_1 = G[(V \setminus S) \cup \{z\}]$ and $G_2 = G'[(V' \setminus S) \cup \{z\}]$. We see that G_2 can be obtained from G_1 by contracting uv. But G_2 must have a cycle, by the minimality of S, so G_1 also has a cycle. Thus, S is minimal in G. $\square$

3. Polynomial Time Approximation Algorithm

In this section we present a polynomial-time algorithm which guarantees an approximation ratio of $n^{2/3}$. As we show in Theorem 4, this ratio is the best that can be hoped for in polynomial time. Later (Theorem 2) we show how to generalize the ideas presented here to obtain an algorithm that achieves a trade-off between the approximation ratio and the (sub-exponential) running time, and show that this trade-off is essentially optimal.

On a high level, our algorithm proceeds as follows: first we identify some easy cases in which applying Lemma 1 or Lemma 2 is *safe*, that is, the value of the optimal solution is guaranteed to stay constant, namely deleting vertices of degree at most 1, and contracting edges between vertices of degree 2. After we apply these reduction rules exhaustively, we compute a minimal fvs S in an arbitrary way. If S is large enough (larger than $n^{1/3}$), we simply return this set.

If not, we apply some counting arguments to show that a vertex $u \in S$ with high degree ($\geq n^{2/3}$) must exist. We then have two cases: either we are able to construct a large minimal fvs just by looking at the neighborhood of u in the forest (and ignoring $S \setminus \{u\}$), or u must share many neighbors with another vertex $v \in S$, in which case we construct a large minimal fvs in the common neighborhood of u, v.

Because our algorithm is constructive (and runs in polynomial time), we find it interesting to remark an interpretation from the point of view of extremal combinatorics, given in Corollary 1.

3.1. Basic Reduction Rules and Combinatorial Tools

We begin by showing two *safe* versions of Lemmas 1, 2.

Lemma 3. *Let G, u be as in Lemma 1 with $d(u) \leq 1$. Then* $\text{mmfvs}(G-u) = \text{mmfvs}(G)$.

Proof. We only need to show that $\text{mmfvs}(G) \leq \text{mmfvs}(G-u)$ (the other direction is given by Lemma 1). Let S be a minimal fvs of G. Then, S is an fvs of $G-u$. Furthermore, $u \notin S$, as S is minimal in G. To see that S is also minimal in $G-u$, note that any cycle of G also exists in $G-u$ (as no cycle contains u). □

Lemma 4. *Let G, u, v be as in Lemma 2 with $d(u) = d(v) = 2$. Then* $\text{mmfvs}(G/uv) = \text{mmfvs}(G)$.

Proof. Let $G' = G/uv$, w be the vertex that replaced u, v in G', and $V' = V(G')$.

We only need to show that $\text{mmfvs}(G) \leq \text{mmfvs}(G')$, as the other direction is given by Lemma 2. Let S be a minimal fvs of G. We consider two cases:

If $u, v \notin S$, then we claim that S is also a minimal fvs of G'. Indeed, $G'[V' \setminus S]$ is obtained from $G[V \setminus S]$ by contracting uv, so both are acyclic. Furthermore, for all $z \in S$, $G'[(V' \setminus S) \cup \{z\}]$ is obtained from $G[(V \setminus S) \cup \{z\}]$ by contracting uv, therefore both have a cycle, hence no vertex of S is redundant in G'.

If $\{u, v\} \cap S \neq \emptyset$, we claim that exactly one of u, v is in S. Indeed, if $u, v \in S$, then $G[(V \setminus S) \cup \{u\}]$ does not contain a cycle going through u, as u has degree 1 in this graph. Without loss of generality, let $u \in S$, $v \notin S$. We set $S' = (S \setminus \{u\}) \cup \{w\}$ and claim that S' is a minimal fvs of G'. It is not hard to see that S' is an fvs of G', since it corresponds to deleting $S \cup \{v\}$ from G. To see that it is minimal, for all $z \in S' \setminus \{w\}$ we observe that $G'[(V' \setminus S') \cup \{z\}]$ obtained from $G'[(V \setminus S) \cup \{z\}]$ by deleting v, which has degree 1. Therefore, this deletion strongly preserves acyclicity. Finally, to see that w is not redundant for S', we observe that $G[(V \setminus S) \cup \{u\}]$ has a cycle, and a corresponding cycle must be present in $G'[(V' \setminus S') \cup \{w\}]$, which is obtained from the former graph by contracting uv. □

Definition 1. *For a graph $G = (V, E)$ we say that G is* reduced *if it is not possible to apply Lemma 3 or Lemma 4 to G.*

We now present a counting argument which will be useful in our algorithm and states, roughly, that if in a reduced graph we find an (not necessarily minimal) fvs, that fvs must have many neighbors in the corresponding forest.

Lemma 5. *Let $G = (V, E)$ be a reduced graph and $S \subseteq V$ a feedback vertex set of G. Let $F = V \setminus S$. Then, $|N(S) \cap F| \geq \frac{|F|}{4}$.*

Proof. Let n_1 be the number of leaves of F, which are vertices with at most one neighbor in F, n_3 the number of vertices of F with at least three neighbors in F, n_{2a} the number of vertices of F with two neighbors in F and at least one neighbor in S, and n_{2b} the number of remaining vertices of F. We have $n_1 + n_{2a} + n_{2b} + n_3 = |F|$. Note also that the vertices counted in n_{2b} have two neighbors in F and no neighbor in S, since the vertices of degree at most one in F are counted in n_1, vertices of degree at least 3 in F are counted in n_3, vertices of degree 2 in F and with at least one neighbor in S are counted in n_{2a}, and G has no isolated vertices since it is reduced.

Claim 1. *In a forest, the number of leaves is greater or equal to the number of vertices of degree at least 3.*

Proof. The average degree in a tree is less than 2. Indeed, we have $\sum_{u \in T} d(u) = 2|E(T)|$, for a tree T. And we know that $|E(T)| \leq n - 1$ since T is a tree. So the average degree in a tree is $(\sum_{u \in T} d(u))/n \leq 2 - 2/n$. Thus, since the average degree in a tree is less than 2, we cannot have more vertices of degree at least 3 than vertices of degree at most 1, and thus the claim follows.

Finally, the same holds for a forest since all connecting components of a forest are trees. □

By the previous Claim, we directly have $n_3 \leq n_1$.

We observe that all isolated vertices of F have a neighbor in S because G do not have any isolated vertices. Furthermore, all leaves of F have a neighbor in S (otherwise we would have applied Lemma 3). This gives $|N(S) \cap F| \geq n_1 + n_{2a}$.

Furthermore, none of the n_{2b} vertices, which have degree two in F and no neighbors in S, can be connected to each other, since then Lemma 4 would apply. Therefore, $n_{2b} \leq n_1 + n_{2a} + n_3$. Indeed, if $n_{2b} > n_1 + n_{2a} + n_3$, then $n_{2b} > |F|/2$, and since these n_{2b} vertices form an independent set, we would have $|E(F)| \geq 2n_{2b} > |F|$, contradicting the assumption that F is a forest.

Putting things together we get $|F| = n_1 + n_{2a} + n_{2b} + n_3 \leq 2n_1 + 2n_{2a} + 2n_3 \leq 4n_1 + 2n_{2a} \leq 4|N(S) \cap F|$. □

We note that Lemma 5 immediately gives an approximation algorithm with ratio $O(\Delta)$.

Lemma 6. *In a reduced graph G with n vertices and maximum degree Δ, every feedback vertex set has size at least $\frac{n}{5\Delta}$.*

Proof. Let S be a feedback vertex set of G and F the corresponding forest. If $|S| < \frac{n}{5\Delta}$ then $|N(S) \cap F| < \frac{n}{5}$ since the maximum degree is Δ. So by Lemma 5, we have $|F| < \frac{4n}{5}$. But then $|V| = |S| + |F| < n$, which is a contradiction. □

Remark 1. *Lemma 5 is tight.*

Proof. Take two copies of a rooted binary tree with n leaves and connect their roots. The resulting tree has $2n$ leaves and $2n - 2$ vertices of degree 3. Subdivide every edge of this tree. Add two vertices u, v connected to every leaf. In the resulting graph $S = \{u, v\}$ is an fvs. The corresponding forest has $8n - 5$ vertices. Indeed, we have: $2n - 3$ new vertices obtained from the subdivisions between the degree-3 vertices; $2n - 2$ vertices of degree 3; and $2(2n)$ leaves and their adjacent new vertices. And we have $2n$ vertices connected to S. The graph is reduced. □

3.2. Polynomial Time Approximation and Extremal Results

We begin with a final intermediate lemma that allows us to construct a large minimal fvs in any reduced graph that is a forest plus one vertex.

Lemma 7. *Let $G = (V, E)$ be a reduced graph and $u \in V$ such that $G - u$ is acyclic. Then it is possible to construct in polynomial time a minimal feedback vertex set S of G with $|S| \geq d(u)/2$.*

Proof. Let $F = V \setminus \{u\}$. Since the graph is reduced, all trees of $G[F]$ contain at least two neighbors of u. Indeed, every tree T of $G[F]$ contains at least two vertices, because otherwise Lemma 3 would apply. Thus every tree T contains at least two leaves, and all leaves must be neighbors of u, because otherwise Lemma 3 would apply.

Now, we edit the graph. As long as there exist $v, w \in F$ with $vw \in E$ and $\{v, w\} \not\subseteq N(u)$, we contract the edge (v, w). Note that we can apply Lemma 2 since v and w do not have any common neighbors (u is not a common neighbor by assumption, and they cannot have a common neighbor in the forest without forming a cycle). This operation does not change $d(u)$, since for two vertices v, w in F that are neighbors of u, the edge vw is not contracted. Therefore, it will be sufficient to construct a minimal fvs in the resulting graph after applying this operation exhaustively, since by Lemma 2 we will be able to construct a minimal fvs in G of the same or greater size.

Suppose now that we have applied this operation exhaustively. We eventually arrive at a graph where u is connected to all vertices of F, since every tree of F initially contain at least two neighbors of u, since all the non-neighbors of u are absorbed by the contraction operation (each contraction decreases $|F \setminus N(u)|$), and since neighbors of u in F are never absorbed by the contraction operation. Therefore, we arrive at a graph with $d(u) = |F'|$ for the new forest F'. And every tree of F' contains at least two vertices.

Now, since $G[F']$ is a forest, it is bipartite, so there is a bipartition $F' = L \cup R$. Without loss of generality, $|L| \leq |R|$. We return the set $S = R$. First, S does have the promised size, since $|S| \geq |F'|/2 = d(u)/2$. Second, S is an fvs, as L is an independent set and $L \cup \{u\}$ is a star. Finally, S is minimal, because every $v \in S$ is connected to u, and also has at least one neighbor $w \in L$ which is also connected to u.

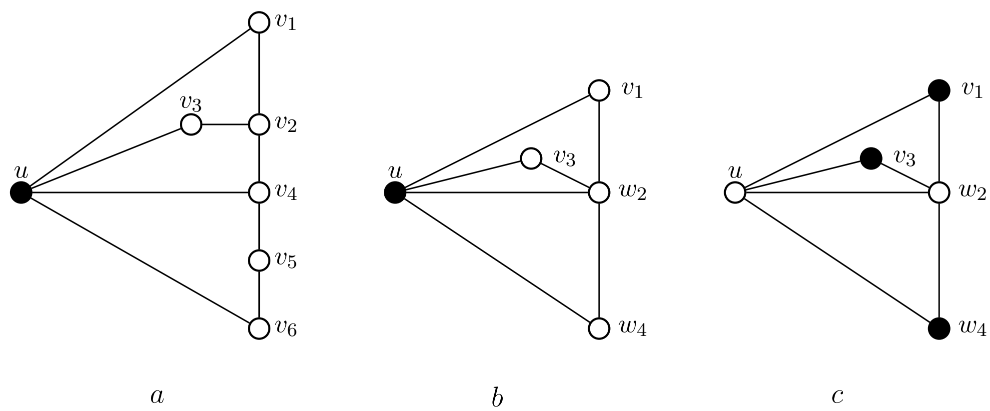

Figure 1: (a) vertex u is a minimal fvs of the given graph and has 4 neighbors in $G[F]$. (b) a contracted form of $G[F]$ with 4 vertices. (c) a new minimal fvs of the result graph of size 3.

An illustration of the process is presented in Figure 1. $\square$

We now present the main result of this section.

Theorem 1. *There is a polynomial time approximation algorithm for* Max Min FVS *with ratio* $O(n^{2/3})$.

Proof. We are given a graph $G = (V, E)$. We begin by applying Lemmas 3 and 4 exhaustively in order to obtain a reduced graph $G' = (V', E')$. Clearly, if we obtain a solution of size at least $|V'|^{1/3}$ in G', since the transformations applied do not change the optimal, and since we can construct a solution of the same size in G (we can construct such a minimal fvs by Lemmas 1 and 2, and it will be of the same size by Lemmas 3 and 4), we get $|V'|^{2/3} \leq |V|^{2/3}$ approximation ratio in G. So, in the remainder, to ease presentation, we assume that G is already reduced and has n vertices.

Our algorithm begins with an arbitrary minimal fvs S. It can be constructed, for example, by starting with $S = V$, and by removing vertices from S until it becomes minimal. If $|S| \geq n^{1/3}$, then we return S. Since the optimal solution cannot have size more than n, we already have a $n^{2/3}$-approximation.

So suppose that $|S| < n^{1/3}$. Let F be the corresponding forest. We have $|F| > n - n^{1/3} > n/2$ for n sufficiently large. By Lemma 5, $|N(S) \cap F| \geq n/8$. Since $|S| < n^{1/3}$, there must exist a vertex $u \in S$ with at least $\frac{|N(S) \cap F|}{|S|} > \frac{n^{2/3}}{8}$ neighbors in F.

Now, let $w \in F \cap N(u)$. We say that w is a *good* neighbor of u if there exists another vertex $w' \in F \cap N(u)$ with $w' \neq w$ and w' is in the same tree of $G[F]$ as w. Otherwise, we say that w is a *bad* neighbor of u. By extension, a tree of $G[F]$ that contains a good (resp. bad) neighbor of u will be called *good* (resp. *bad*). Note that every vertex of $N(u) \cap F$ is either good or bad.

Recall that $|N(u) \cap F| \geq \frac{n^{2/3}}{8}$. We distinguish between the following two cases: either u has at least $\frac{n^{2/3}}{16}$ good neighbors in F, or it has at least that many bad neighbors in F.

In the former case, we delete from the graph the set $S \setminus \{u\}$, and apply Lemmas 3 and 4 exhaustively again. We claim that the number of good neighbors of u does not decrease in this process. Indeed, two good neighbors of u cannot be contracting using Lemma 4, since they have a common neighbor, namely u. Furthermore, suppose w is the first good neighbor of u to be deleted using Lemma 3. This would mean that w currently has no other neighbor except u. However, since w is good, there initially was a vertex $w' \in N(u)$ in the same tree of $G[F]$ as w. And since w' has not been deleted yet, since we assumed that w was the first to be deleted, and since Lemmas 3 and 4 cannot disconnect two vertices which are in the same component, we obtain that the vertex w cannot be removed by Lemma 3. Thus, we have a reduced graph, where $\{u\}$ is an fvs, and with $d(u) \geq \frac{n^{2/3}}{16}$. So, by Lemma 7, we obtain a minimal fvs of size at least $\frac{n^{2/3}}{32}$, which is an $O(n^{1/3})$-approximation.

In the latter case, u has at least $\frac{n^{2/3}}{16}$ bad neighbors in F. Consider such a bad tree T. The tree T must have a neighbor in $S \setminus \{u\}$. Indeed, if $|T| = 1$, then the vertex in T must have another neighbor in S, because otherwise it should have been deleted by Lemma 3. And if $|T| \geq 2$, then one vertex is a neighbor of u and at least one leaf is connected to S, because otherwise this leaf should have been deleted by Lemma 3. Furthermore, since u is connected to one vertex in each bad tree, u is connected to at least $\frac{n^{2/3}}{16}$ bad trees. We now find a vertex $v \in S \setminus \{u\}$ such that v is connected to the maximum number of bad trees connected to u. Since $|S| < n^{1/3}$, v must be connected to at least $\frac{n^{2/3}}{16|S|} \geq \frac{n^{1/3}}{16}$ bad trees connected to u.

Now, we delete from the graph the set $S \setminus \{u, v\}$ as well as all trees of $G[F]$, except the bad trees connected to u and v. Consider such a bad tree T connected to both u and v, and let $u' \in T \cap N(u)$ and $v' \in T \cap N(v)$ such that u' and v' are as close as possible in T (note that perhaps $u' = v'$). We delete all vertices of the tree T except those on the path from u' to v', and then we contract all internal edges of this path (note that internal vertices of this path are not connected to u and v by the selection of u', v'). By Lemma 1 and 2, if we are able to produce a large minimal fvs in the resulting graph, we obtain a solution for G, since we have applied these two Lemmas to obtain the resulting graph. We have that in the resulting graph, every bad tree T connected to u and v has been reduced to a single vertex connected to u and v. So the graph is either a $K_{2,s}$ with $s \geq \frac{n^{1/3}}{16}$, or the same graph with the addition of the edge uv. In either case, by starting with the fvs that contains all vertices except u and v, and making it minimal, we obtain a solution of size at least $s-1$, which gives a $O(n^{2/3})$-approximation. □

Corollary 1. *For any reduced graph G on n vertices we have* $\text{mmfvs}(G) = \Omega(n^{1/3})$.

Proof. We simply note that the algorithm of Theorem 1 always constructs a solution of size at least $\frac{n^{1/3}}{c}$, where c is a small constant, assuming that the original n-vertex graph G was reduced. □

Remark 2. *Corollary 1 is tight.*

Proof. Take a K_n and for every pair of vertices u, v in the clique, add $2n$ new vertices connected only to u and v. The graph has order $n + 2n\binom{n}{2} = n + n^2(n-1) = n^3 - n^2 + n \geq n^3/2$ for n sufficiently large. Any minimal fvs of this graph must contain at least $n-2$ vertices of the clique. As a result its maximum size is at most $n - 2 + 2n \leq 3n$. We have $\text{mmfvs}(G) \leq 3n$ so $\text{mmfvs}(G) = O(|V(G)|^{1/3})$. □

Theorem 1 also implies the existence of a cubic kernel of Max Min FVS when parameterized by the solution size k. Recall that the reduction rules do not change the solution size. We suppose that the reduced graph has n vertices. For a small constant c, if $n \geq c^3k^3$, then we can always produce a solution of size at least $n^{1/3}/c = k$, and thus the answer is YES. Otherwise, we have a cubic kernel.

Corollary 2. Max Min FVS *admits a cubic kernel when parameterized by the solution size.*

Finally, we remark that a similar combinatorial point of view can be taken for the related problem of Max Min VC, giving another intuitive explanation for the difference in approximability between the two problems.

Remark 3. *Any graph $G = (V, E)$ without isolated vertices, has a minimal vertex cover of size at least $\sqrt{|V|}$, and this is asymptotically tight.*

Proof. We will prove the statement under the assumption that G is connected. If not, we can treat each component separately. If the components of G have sizes $n_1, \ldots, n_k$, then we rely on the fact that $\sum_{i=1}^{k} \sqrt{n_i} \geq \sqrt{\sum_{i=1}^{k} n_i}$ and that the union of the minimal vertex covers of each component is a minimal vertex cover of G.

If $G = (V, E)$ has a vertex u of degree at least $\sqrt{n}$, then we begin with the vertex cover $V \setminus \{u\}$ and remove vertices until it becomes minimal. In the end, our solution contains a superset of $N(u)$, therefore we have a minimal vertex cover of size at least $\sqrt{n}$ as promised. If, on the other hand, $\Delta(G) < \sqrt{n}$, then any vertex cover of G must have size at least $\sqrt{n}$. Indeed, a vertex cover of size at most $\sqrt{n} - 1$ can cover at most $(\sqrt{n} - 1)\sqrt{n} < n - 1$ edges, but since G is connected we have $|E(G)| \geq n - 1$. So, in this case, any minimal vertex cover has the promised size.

To see that the bound given is tight, take a K_n and attach n leaves to each of its vertices. This graph has $n^2 + n$ vertices, but any minimal vertex cover has size at most $(n-1) + n = 2n - 1$. □

4. Sub-exponential Time Approximation

In this section we give an approximation algorithm that generalizes our $n^{2/3}$-approximation and is able to guarantee any desired performance, at the cost of increased running time. On a high level, our initial approach again constructs an arbitrary minimal fvs S and if S is clearly large enough, returns it. However, things become more complicated from then on, as it is no longer sufficient to consider vertices of S individually or in pairs. We therefore need several new ideas, one of which is given in the following lemma, which states that we can find a constant factor approximation in time exponential in the size of a given fvs. This will be useful as we will use the assumption that S is "small" and then cut it up into even smaller pieces to allow us to use Lemma 8.

Lemma 8. *Given a graph $G = (V, E)$ on n vertices and a feedback vertex set $S_0 \subseteq V$ of size k, it is possible to produce a minimal fvs S' of G of size $|S'| \geq \frac{\text{mmfvs}(G)}{3}$ in time $n^{O(k)}$.*

Before we prove this Lemma, let us point out that for $k = 1$, Max Min FVS can be solved optimally in time $O(n)$, using standard arguments from parameterized complexity. Indeed, in this case, the graph G has treewidth 2, so by invoking Courcelle's Theorem and since the properties "S is an fvs" and "S is minimal" are MSO-expressible [15], we can solve the problem optimally in time $O(n)$. Unfortunately, this type of argument is not good enough for larger value of k, as the running time guaranteed by Courcelle's Theorem could depend super-exponentially on k. We could try to avoid this by formulating a treewidth-based dynamic programming algorithm to obtain a better running time, but we prefer to give a simpler more direct branching algorithm, since this is good enough for the super-polynomial approximation algorithm we seek to design.

Proof. We will assume that S_0 is minimal, because otherwise we can remove vertices from it to make it minimal, and this only decreases the running time of our algorithm. As a result, we assume also that $\mathrm{mmfvs}(G) \geq 3k$, as otherwise S_0 is already a 3-approximation.

Let S^* be a maximum minimal fvs in G, and let $F^* = V \setminus S^*$. We formulate an algorithm that maintains two disjoint sets of vertices S and F, which, intuitively, correspond to the vertices we have decided to place in the fvs or in the induced forest, respectively. We will denote $U = V \setminus (S \cup F)$ the set of "undecided" vertices. Our algorithm will sometimes "guess' some vertices of U to be placed in S or F, and we will upper-bound the guessing possibilities by $n^{O(k)}$.

Throughout the algorithm, we will work to maintain the following four invariants:

1. $S \cup F$ is an fvs of G;
2. $S \subseteq S^*$ and $F \subseteq F^*$;
3. $G[F]$ is acyclic and has at most $2k$ components;
4. All vertices of S have at least two neighbors in F.

The algorithm consists of the following five steps.

Step 1. We are guessing a set $F_0 \subseteq S_0$ such that $G[F_0]$ is acyclic and we set $F = F_0$ and $S = S_0 \setminus F_0$. Then, if there exists a vertex $u \in S$ which does not satisfy Property 4, we guess one or two vertices from $N(u) \cap U$ and place them into F, so that u has indeed two neighbors in F. We continue in this way until Property 4 is satisfied for all vertices of S.

Step 2. Now, we need to define the notion of "connector". Formally, a connector is a path $V(P) \subseteq F^* \setminus F$ such that $G[F \cup P]$ has strictly fewer components than $G[F]$. Our algorithm will now repeatedly guess if a connector exists, and if it does it will guess the first and last vertices u and v of P. Then, we set $F = F \cup P$, and we continue guessing until we guess that no connector exists.

Step 3. We consider every vertex $u \in U$ that has at least two neighbors in F and place all such vertices in S. We are now in a situation where every vertex of U has at most one neighbor in F.

Step 4. We construct a new graph H by deleting from G all of S and replacing F by a single vertex f that is connected to $N(F) \cap U$. Note that H is a simple graph, i.e. it has no parallel edges, because otherwise a vertex of U would have two neighbors in F, and we have put all these vertices of U in S in the previous step. Moreover, H has an fvs of size 1, namely the set $\{f\}$. We therefore use the aforementioned algorithm implied by Courcelle's Theorem to produce a maximum minimal fvs of H, which, without loss of generality, does not contain f. Let $S_H \subseteq U$ be this fvs.

Step 5. Finally, in G, the set $S \cup S_H$ is an fvs. But it might be not minimal, so we remove vertices from it until it is minimal. Let S' be this minimal fvs obtained.

Now let us prove that in each step of the algorithm, if we have made the correct guesses, then the sets S and F satisfy all the four properties. Furthermore, we will prove that the number of guesses in each step are bounded by $n^{O(k)}$.

In the first half of step one, Property 1 is satisfied as $S \cup F = S_0$ is an fvs of G and Property 2 is satisfied for the right guess $F_0 = F \cap S_0$. In the second half of step one we add vertices in F until Property 4 is satisfied for all vertices of S. Observe that any $u \in S$ has a private cycle in $G[F^* \cup \{u\}]$ so if we have made correct guesses all the properties 1, 2 and 4 must be satisfied. Last, because we have added at most 2 vertices for each vertex of S, it follows that F contains at most $2k$ vertices, hence at most $2k$ components, so Property 3 is also satisfied. So far, the total running time is upper-bounded by $2^k n^{2k}$: 2^k for guessing $F_0 \subseteq S_0$ and n^{2k} for guessing at most two neighbors for every $u \in S$.

In the second step we are guessing the first and last vertices u and v of a connector P. Note that $u, v \in U$, and if we rightfully guess u and v, then we can infer all of P, since $G[U]$ is acyclic and there is at most one path from u to v in $G[U]$. Note that guessing the two endpoints of a connector gives n^2 possibilities, and that adding a connector to F decreases the number of connected components of F by at least one, which can happen at most $2k$ times by Property 3. So the total running time of this procedure is upper-bounded by $n^{O(k)}$. We now show that the four properties are satisfied so far. Property 1 is satisfied since $S \cup F$ was already a fvs of G before adding the connectors to F. For the right guess of u and v of a connecter P, $V(P) \subseteq F^* \setminus F$ holds, which implies that adding the vertices of $V(P)$ to F preserves Property 2. Property 3 is satisfied since adding a connector decreases the number of components by at least one. And Property 4 is satisfied since every vertex of S already had two neighbors in F before adding the connectors.

In the third step, it is easy to see that Properties 1, 3 and 4 are still satisfied. Furthermore, if our guesses so far are correct, all vertices $u \in U$ such that u has at least two neighbors in F belong to S^*. Indeed, they have at least two neighbors in F which are connected to each other, because otherwise they would function as connectors in F, and we assume that we have correctly guessed that no more connectors exist. Thus, these vertices u must be in S^* in order to dominate the cycle that go through their two neighbors in F.

In the fourth and fifth steps we do not change the sets S and F any more. Therefore, we only need to prove that this solution S' is a 3-approximation. To see that the resulting solution has the desired size, we focus on the case where all guesses were correct, and therefore where Properties 1-4 were maintained throughout the execution of the algorithm. As mentioned earlier, the total running time of this algorithm is $n^{O(k)}$.

We first observe that $\text{mmfvs}(H) \geq \text{mmfvs}(G) - |S|$. Indeed, the set $S_1 = S^* \setminus S$ is a minimal fvs of H. To see that S_1 is a fvs, suppose that H contains a cycle after deleting S_1. This cycle must necessarily go through f, since $G[U]$ is acyclic. Now, let P be the vertices of this cycle except f. We have $P \subseteq U \setminus S^*$, so $P \subseteq F^*$. However, this means that either P forms a cycle with a component of F, which contradicts the acyclicity of F^* by Property 2, or P is a connector, which contradicts our guess that no other connector exists. Therefore, we obtain a contradiction, and S_1 must be an fvs of H. To see that it is minimal, we note that for every $u \in S_1$, there is a private cycle in $G[U \cup F \cup \{u\}]$, since $S_1 = S^* \setminus S$ and $S \subseteq S^*$ by Property 2. And this private cycle is not destroyed by contracting the vertices of F into f, since $F \subseteq F^*$ by Property 2.

We now have that $|S_H \cup S| \geq |S^*|$, because $|S_H| \geq |S^* \setminus S|$. We argue that in the process of making S_H minimal to obtain S', we delete at most $2k$ vertices. Indeed, every time a vertex u of S is removed from $S \cup S_H$ as redundant, since u has at least two neighbors in F by Property 4, the number of components of $G[F]$ must decrease. Similarly, every time a vertex $u \in S_H$ is removed as redundant, consider the private cycle of u in $H \setminus S_H$. All of the vertices of this cycle are present in G after we remove S_H, except f. Therefore, this cycle must form a path between two distinct components of $G[F]$, since $G[U]$ is acyclic, and because u has been considered redundant if its private cycle in $G[H]$ does not exist in G, thus if this cycle forms a path with two distinct components of $G[F]$. We conclude that, since removing a vertex from $S \cup S_H$ decreases the number of components in $G[F]$, and since they are at most $2k$ such components in $G[F]$ by Property 3, we have $|S'| \geq |S^*| - 2k$. But recall that we have assumed $k \leq \frac{|S^*|}{3}$, so we obtain $|S'| \geq \frac{|S^*|}{3}$. $\square$

We now present the main result of this section.

Theorem 2. *There is an algorithm which, given an n-vertex graph $G = (V, E)$ and a value r, produces an r-approximation for* Max Min FVS *in G in time $n^{O(n/r^{3/2})}$.*

Proof. First, let us note that we may assume that r is $\omega(1)$, because if r is bounded above by a constant, then we can solve the problem exactly in the given time. To ease presentation, we will give an algorithm with approximation ratio $O(r)$. A ratio of approximation ratio exactly r can be obtained by multiplying r with an appropriate small constant.

Our algorithm borrows several of the basic ideas from Theorem 1, but requires some new ingredient, including the algorithm of Lemma 8. The first step is, again, to construct a minimal fvs S in some arbitrary way, for example by setting $S = V$ and then removing vertices from S until it becomes minimal. If $|S| \geq n/r$, then we already have an r-approximation, so in this case we simply return S. So we assume that $|S| < n/r$. From this point, this algorithm departs from the algorithm of Theorem 1, because it is no longer sufficient to compare the size of the output solution with a function of n, we need to compare it to the actual optimal value in order to obtain a ratio of r.

Let us now present our algorithm. Let $k = \lceil \sqrt{r} \rceil$. Partition S into k almost equal-sized parts $S_1, \ldots, S_k$. Our algorithm proceeds as follows: for each $i, j \in \{1, \ldots, k\}$ with i and j not necessarily distinct, consider the graph $G_{i,j}$ obtained by deleting all vertices of $S \setminus (S_i \cup S_j)$. Compute, using the algorithm of Lemma 8, a solution for $G_{i,j}$, taking into account that $S_i \cup S_j$ is a fvs of G_{ij}, though not necessarily minimal. Then, for each of the solutions found, extend it to a solution of G using Lemma 1. Finally, output the largest solution encountered.

The algorithm runs in the promised time: we have $|S_i \cup S_j| < \frac{2n}{rk}$, so the algorithm of Lemma 8 runs in time $n^{O(n/r^{3/2})}$, and the rest of the algorithm runs in polynomial time.

Let us now analyze the approximation ratio of the produced solution. Let S^* be an optimal solution, and let $F = F \setminus S$ and $F^* = V \setminus S^*$ be the induced forests corresponding to S and S^*, respectively. We would like to argue that one of the considered sub-problems contains at least $1/r$ fraction of S^*, and that most of these vertices form part of a minimal fvs of that subgraph.

We will define the notion of "type" for every $u \in S^* \cap F$. For each such u, there must exist a private cycle in the graph $G[F^* \cup \{u\}]$, since S^* is a minimal fvs. Call this cycle $c(u)$, and consider one such cycle if several exist. The cycle $c(u)$ must intersect with S since S is an fvs. So let v be the vertex of $c(u) \cap S$ closest to u on this cycle, and let v' be the vertex of $c(u) \cap S$ closest to u if we traverse the cycle in the opposite direction. Note that perhaps $v = v'$. Suppose that $v \in S_i$ and $v' \in S_j$, and without loss of generality, $i \leq j$. We then say that $u \in S^* \cap F$ has type (i, j). In this way, we define a type of every $u \in S^* \cap F$. Note that, according to our definition, all internal vertices of the paths in $c(u)$ from u to v and from u to v' belong to $F^* \cap F$.

According to the definition of the previous paragraph, there are $\binom{k}{2}+k = k(k+1)/2 \leq r$ possible types of vertices in $S^* \cap F$. Therefore, there must exist a type (i, j) such that at least $\frac{|S^* \cap F|}{r}$ vertices have this type. We now concentrate on the corresponding graph $G_{i,j}$, for the type (i, j) that satisfies this condition. Our algorithm has constructed $G_{i,j}$ be deleting all vertices of $S \setminus (S_i \cup S_j)$. We will prove that this graph has a minimal feedback vertex set of size comparable to $\frac{|S^* \cap F|}{r}$.

For the sake of the analysis, construct a minimal feedback vertex set $S_{i,j}$ of $G_{i,j}$ as follows: start with the fvs $S_{i,j} = S^* \cap (F \cup S_i \cup S_j)$. Let $F_{i,j}$ be the corresponding induced forest $F_{i,j} = F^* \cap (F \cup S_i \cup S_j)$. The set $S_{i,j}$ is a feedback vertex set of $G_{i,j}$ as it contains all vertices of S^* found in $G_{i,j}$ and S^* is a feasible fvs of all of G. We then make $S_{i,j}$ minimal by removing vertices from it until it becomes minimal. Call the resulting set $S'_{i,j} \subseteq S_{i,j}$ and the corresponding induced forest $F'_{i,j} \supseteq F_{i,j}$.

We will prove now that the number of vertices of $S^* \cap F$ of type (i, j) which have been deleted in the process of making $S_{i,j}$ minimal is upper-bounded by $|S_i \cup S_j|$. Consider such a vertex $u \in (S_{i,j} \cap F) \setminus S'_{i,j}$ of type (i, j), and let $c(u)$ be the cycle that defines the type of u, and v, v' be the vertices of $S_i \cup S_j$ which are closest to u on the cycle in either direction. As we have mentioned earlier, all vertices of $c(u)$ in the paths from u to v and from u to v' belong to $F^* \cap F$ and therefore to $F_{i,j}$. If u was removed as redundant, this means that v and v' must have been in distinct connected components at the moment u was removed from the fvs $S_{i,j}$, because since $c(u)$ is a private cycle of u, if u has been removed, it means that v and v' are only connected by a path going through u and no other vertex. However, the addition of u to the induced forest creates a path from v to v' in the induced forest, and hence decreases the number of connected components containing vertices of $S_i \cup S_j$. The number of such connected components cannot decrease more than $|S_i \cup S_j|$ times. Thus, in the process of making $S_{i,j}$ minimal, we have removed at most $|S_i \cup S_j|$ vertices of type (i, j) from $S_{i,j} \cap F$.

Using the above analysis, and the assumption that $S_{i,j}$ contains at least $\frac{|S^* \cap F|}{r}$ vertices of type (i, j), we have that $\text{mmfvs}(G_{i,j}) \geq |S'_{i,j}| \geq \frac{|S^* \cap F|}{r} - |S_i \cup S_j|$. Now, we can assume that $|S^* \cap S| < \frac{|S^*|}{r}$, because otherwise S is already an r-approximation. So we can assume that $|S^* \cap F| \geq \frac{(r-1)|S^*|}{r}$. Furthermore, we obtain $|S_i \cup S_j| \leq \frac{2|S|}{\sqrt{r}} \leq \frac{2|S^*|}{r\sqrt{r}}$, where again we assume that S is not already an r-approximation. Putting things together, we obtain $\text{mmfvs}(G_{i,j}) \geq \frac{(r-1)|S^*|}{r^2} - \frac{2|S^*|}{r\sqrt{r}} \geq \frac{|S^*|}{r}$, for r sufficiently large. Hence, since our algorithm will return a solution that is at least as large as $\frac{\text{mmfvs}(G_{i,j})}{3}$, we obtain an $O(r)$-approximation. □

5. Hardness of Approximation and NP-hardness

In this section we establish lower bound results showing that the approximation algorithms given in Theorems 1 and 2 are essentially optimal, under standard complexity assumptions.

5.1. Hardness of Approximation in Polynomial Time

We begin by showing that the best approximation ratio achievable in polynomial time is indeed (essentially) $n^{2/3}$. For this, we rely on the celebrated result of Håstad on the hardness of approximating Max Independent Set, which was later derandomized by Zuckerman, cited below.

Theorem 3. *[29, 42] For any $\epsilon > 0$, there is no polynomial time algorithm which approximates* Max Independent Set *with a ratio of $n^{1-\epsilon}$, unless P = NP.*

Starting from this result, we present a reduction to Max Min FVS.

Theorem 4. *For any $\epsilon > 0$,* Max Min FVS *is inapproximable within a factor of $n^{2/3-\epsilon}$ unless P = NP.*

Proof. We give a gap-preserving reduction from Max Independent Set, which cannot be approximated within a factor of $n^{1-\epsilon}$, unless P = NP. We are given a graph $G = (V, E)$ on n vertices as an instance of Max

Independent Set. Recall that $\alpha(G)$ denotes the size of the maximum independent set of G.

We transform G into an instance of Max Min FVS as follows: For every pair of $u, v \in V$, we add n vertices such that they are adjacent only to u and v. We denote by I_{uv} the set of such vertices. Then I_{uv} is an independent set. Let $G' = (V', E')$ be the constructed graph.

We now make the following two claims:

Claim 2. $\mathrm{mmfvs}(G') \geq (n-1)\binom{\alpha(G)}{2}$

Proof. We construct a minimal fvs of G' as follows: let C be a minimum vertex cover of G. Then we begin with the set that contains C and the union of all I_{uv} (which is clearly an fvs) and remove vertices from it until it becomes minimal. Let S be the final minimal fvs. We observe that for all $u, v \in V \setminus C$, S contains at least $n-1$ of the vertices of I_{uv}. Since C is a minimum vertex cover of G, there are $\binom{\alpha(G)}{2}$ pairs $u, v \in V \setminus C$. ☐

Claim 3. $\mathrm{mmfvs}(G') \leq n\binom{2\alpha(G)}{2} + n$

Proof. Let S be a minimal fvs of G' and F be the corresponding forest. It suffices to show that $|S \setminus V| \leq n\binom{2\alpha(G)}{2}$, since $|S \cap V| \leq n$. Consider now a set I_{uv}. If $u \in S$ or $v \in S$, then $I_{uv} \cap S = \emptyset$, because all vertices of I_{uv} have at most one neighbor in F, and are therefore redundant. So, I_{uv} contains (at most n) vertices of S only if $u, v \in F$. However, $|F \cap V| \leq 2\alpha(G)$, because F is bipartite, so $F \cap V$ induces two independent sets, both of which must be at most equal to the maximum independent set of G. So the number of pairs $u, v \in F \cap V$ is at most $\binom{2\alpha(G)}{2}$ and since each corresponding I_{uv} has size n, we get the promised bound. ☐

The two claims together imply that there exist constants c_1, c_2 such that (for sufficiently large n) we have $c_1 n(\alpha(G))^2 \leq \mathrm{mmfvs}(G') \leq c_2 n(\alpha(G))^2$. That is, $\mathrm{mmfvs}(G') = \Theta(n(\alpha(G))^2)$.

Suppose now that there exists a polynomial-time approximation algorithm which, given a graph G', produces a minimal fvs S with the property $\frac{\mathrm{mmfvs}(G')}{r} \leq |S| \leq \mathrm{mmfvs}(G')$, that is, there exists an r-approximation for Max Min FVS. Running this algorithm on the instance we constructed, we obtain that $\frac{c_1 n(\alpha(G))^2}{r} \leq |S| \leq c_2 n(\alpha(G))^2$. Therefore, $\frac{\alpha(G)}{\sqrt{rc_2/c_1}} \leq \sqrt{\frac{|S|}{c_2 n}} \leq \alpha(G)$. As a result, we obtain an $O(\sqrt{r})$ approximation for the value of $\alpha(G)$. We therefore conclude that, unless P = NP, any such algorithm must have $\sqrt{r} > n^{1-\epsilon}$, for any $\epsilon > 0$, hence, $r > n^{2-\epsilon}$, for any $\epsilon > 0$. Since the graph

G' has $N = \Theta(n^3)$ vertices, we get that no approximation algorithm can achieve a ratio of $N^{2/3-\epsilon}$. □

We notice that in the construction of the previous theorem, the maximum degree of the graph is approximately equal to the approximation gap. Thus, the following corollary also holds.

Corollary 3. *For any positive constant ϵ,* Max Min FVS *is inapproximable within a factor of $\Delta^{1-\epsilon}$ unless P = NP.*

5.2. Hardness of Approximation in Sub-Exponential Time

In this section we extend Theorem 4 to the realm of sub-exponential time algorithms. We recall the following result of Chalermsook et al.:

Theorem 5. *[12] For any $\epsilon > 0$ and any sufficiently large r, if there exists an r-approximation algorithm for* Max Independent Set *running in $2^{(n/r)^{1-\epsilon}}$, then the randomized ETH is false.*

We remark that Theorem 5, which gives an almost tight running time lower bound for Max Independent Set, has already been used as a starting point to derive a similarly tight bound for the running time of any sub-exponential time approximation for Max Min VC. Here, we modify the proof of Theorem 4 to obtain a similarly tight result for Max Min FVS. Nevertheless, the reduction for Max Min FVS is significantly more challenging, because the ideas used in Theorem 4 involve an inherent quadratic (in n) blow-up of the size of the instance. As a result, in addition to executing an appropriately modified version of the reduction of Theorem 4, we are forced to add an extra "sparsification" step, and use a probabilistic analysis with Chernoff bounds to argue that this step does not destroy the inapproximability gap.

Theorem 6. *For any $\epsilon > 0$ and any sufficiently large r, if there exists an r-approximation algorithm for* Max Min FVS *running in $2^{(n/r^{3/2})^{1-\epsilon}}$, then the randomized ETH is false.*

Proof. We recall some details about the reduction used to prove Theorem 5. The reduction of [12] begins from a 3-SAT instance ϕ on n variables, and for any ϵ, r, constructs a graph G with $n^{1+\epsilon}r^{1+\epsilon}$ vertices which (with high probability) satisfies the following properties: if ϕ is satisfiable, then $\alpha(G) \geq n^{1+\epsilon}r$; otherwise $\alpha(G) \leq n^{1+\epsilon}r^{2\epsilon}$. Hence, any approximation algorithm with ratio $r^{1-2\epsilon}$ for Max Independent Set would be able to distinguish between the two cases (and solve the initial 3-SAT instance). If, furthermore, this algorithm runs in $2^{(|V|/r)^{1-2\epsilon}}$, we get a sub-exponential algorithm for 3-SAT.

Suppose we are given ϵ, r, and we want to prove the claimed lower bound on the running time of any algorithm that r-approximates Max Min FVS. To ease presentation, we will assume that r is the square of an integer (this can be achieved without changing the value of r by more than a small constant). We will also perform a reduction from 3-SAT to show that an algorithm that achieves this ratio too rapidly would give a sub-exponential (randomized) algorithm for 3-SAT. We begin by executing the reduction of [12], starting from a 3-SAT instance ϕ on n variables, but adjusting their parameter r appropriately so we obtain a graph G with the following properties (with high probability):

- $|V(G)| = n^{1+\epsilon}r^{1/2+\epsilon}$
- If ϕ is satisfiable, then $\alpha(G) \geq n^{1+\epsilon}r^{1/2}$
- If ϕ is not satisfiable, then $\alpha(G) \leq n^{1+\epsilon}r^{2\epsilon}$

We now construct a graph G' as follows: for each pair $u, v \in V(G)$, we introduce an independent set I_{uv} of size $\sqrt{r}$ connected to u, v. We claim that G' has the following properties (assuming G has the properties cited above):

- $|V(G')| = \Theta(n^{2+2\epsilon}r^{3/2+2\epsilon})$
- If ϕ is satisfiable, then $\text{mmfvs}(G') = \Omega(n^{2+2\epsilon}r^{3/2})$
- If ϕ is not satisfiable, then $\text{mmfvs}(G') = O(n^{2+2\epsilon}r^{1/2+4\epsilon})$

Before proceeding, let us establish the properties mentioned above. The size of $|V(G')|$ is easy to bound, as for each of the $\binom{|V(G)|}{2}$ pairs of vertices of G we have constructed an independent set of size $\sqrt{r}$. If ϕ is satisfiable, we construct a minimal fvs of G' by starting with a minimum vertex cover C of G to which we add all vertices of all I_{uv}. We then make this fvs minimal. We claim that for each I_{uv} for which $u, v \in V \setminus C$, our set will in the end contain all of I_{uv}, except maybe at most one vertex. Furthermore, if one vertex of I_{uv} is removed from the fvs as redundant, this decreases the number of components of the induced forest that contain vertices of V (as u, v are now in the same component). This cannot happen more than $|V(G)|$ times. The number of I_{uv} with $u, v \in V \setminus C$ is $\binom{\alpha(G)}{2} = \Omega(n^{2+2\epsilon}r)$. So, $\text{mmfvs}(G') = \Omega(n^{2+2\epsilon}r^{3/2} - |V(G)|)$.

For the third property, take any minimal fvs S of G' and let F be the corresponding forest. We have $|F \cap V| \leq 2\alpha(G)$, because F is bipartite. It is sufficient to bound $|S \setminus V|$ to obtain the bound (as $|S \cap V|$ is already small enough). To do this, we note that in a set I_{uv} where u, v are not both in F, we have $I_{uv} \cap S = \emptyset$, as all vertices of I_{uv} are redundant. So, the number of sets I_{uv} which contribute vertices to S is at most $\binom{|F \cap V|}{2} = O(n^{2+2\epsilon} r^{4\epsilon})$. Each such set has size $\sqrt{r}$, giving the claimed bound.

We have now constructed an instance where the gap between the values for mmfvs(G'), depending on whether ϕ is satisfiable, is almost r (in fact, it is $r^{1-4\epsilon}$, but we can make it equal to r by adjusting the parameters accordingly). The problem is that the order of the new graph depends quadratically on n. This blow-up makes it impossible to obtain a running time lower bound, as a fast approximation algorithm for Max Min FVS (say with running time $2^{n/r^2}$) would not result in a sub-exponential algorithm for 3-SAT. We therefore need to "sparsify" our instance.

We construct a graph G'' by taking G' and deleting every vertex of $V(G') \setminus V(G)$ with probability $\frac{n-1}{n}$. That is, every vertex of the independent sets I_{uv} we added survives (independently) with probability $1/n$. We now claim the following properties hold with high probability:

- $|V(G'')| = \Theta(n^{1+2\epsilon} r^{3/2+2\epsilon})$
- If ϕ is satisfiable, then mmfvs$(G'') = \Omega(n^{1+2\epsilon} r^{3/2})$
- If ϕ is not satisfiable, then mmfvs$(G'') = O(n^{1+2\epsilon} r^{1/2+4\epsilon})$

Before we proceed, let us explain why if we establish that G'' satisfies these properties, then we obtain the theorem. Indeed, suppose that for some sufficiently large r and $\epsilon > 0$, there exists an approximation algorithm for Max Min FVS with ratio $r^{1-5\epsilon}$ running in time $2^{(N/r^{3/2})^{1-10\epsilon}}$ for graphs with N vertices. The algorithm has sufficiently small ratio to distinguish between the two cases in our constructed graph G'', as the ratio between mmfvs(G'') when ϕ is satisfiable or not is $\Omega(r^{1-4\epsilon})$ (and r is sufficiently large), so we can use the approximation algorithm to solve 3-SAT. Furthermore, to compute the running time we see that $N/r^{3/2} = \Theta(n^{1+2\epsilon} r^{2\epsilon}) = O(n^{1+4\epsilon})$. Therefore, $(N/r^{3/2})^{1-10\epsilon} = o(n)$ and we get a sub-exponential time algorithm for 3-SAT. We conclude that for any sufficiently large r and any $\epsilon > 0$, no algorithm achieves ratio $r^{1-5\epsilon}$ in time $2^{(N/r^{3/2})^{1-10\epsilon}}$. By adjusting r, ϵ appropriately we get the statement of the theorem.

Let us therefore try to establish that the three claimed properties all hold with high probability. We will use the following standard Chernoff bound:

suppose $X = \sum_{i=1}^{n} X_i$ is the sum of n independent random 0/1 variables X_i and that $E[X] = \sum_{i=1}^{n} E[X_i] = \mu$. Then, for all $\delta \in (0,1)$ we have $Pr[|X - \mu| \geq \delta\mu] \leq 2e^{-\mu\delta^2/3}$.

The first property is easy to establish: we define a random variable X_i for each vertex of each I_{uv} of G'. This variable takes value 1 if the corresponding vertex appears in G'' and 0 otherwise. Let X be the sum of the X_i variables, which corresponds to the number of such vertices appearing in G''. Suppose that the number of vertices in sets I_{uv} of G' is $cn^{2+2\epsilon}r^{3/2+2\epsilon}$, where c is a constant. Then, $E[X] = cn^{1+2\epsilon}r^{3/2+2\epsilon}$. Also, $Pr[|X - E[X]| \geq \frac{E[X]}{2}] \leq 2e^{-E[X]/12} = o(1)$. So with high probability, $|V(G'')|$ is of the promised magnitude.

The second property is also straightforward. This time we consider a maximum minimal fvs S of G' of size $cn^{2+2\epsilon}r^{3/2}$. Again, we define an indicator variable for each vertex of this set in sets I_{uv}. The expected number of such vertices that survive in G'' is $cn^{1+2\epsilon}r^{3/2}$. As in the previous paragraph, with high probability the actual number will be close to this bound. We now need to argue that (almost) the same set is a minimal fvs of G''. We start in G'' with (the surviving vertices of) S, which is clearly an fvs of G'', and delete vertices until the set is minimal. We claim that the size of the set will decrease by at most $|V(G)| = n^{1+\epsilon}r^{1+\epsilon}$. Indeed, if $S \cap I_{uv} \neq \emptyset$, then $u, v \notin S$. The two vertices u, v are (deterministically) included in G'' and start out in the corresponding induced forest in our solution. If a vertex of $S \cap I_{uv}$ is deleted as redundant, placing that vertex in the forest will put u, v in the same component, reducing the number of components of the forest with vertices from $|V(G)|$. This can happen at most $|V(G)|$ times. Since $|V(G)| < \frac{c}{10}(n^{1+2\epsilon}r^{3/2})$ (for n, r sufficiently large), deleting these redundant vertices will not change the order of magnitude of the solution.

Finally, in order to establish the third property we need to consider every possible minimal fvs of G'' and show that none of them end up being too large. Consider a set $F \subseteq V(G)$ that induces a forest in G. Our goal is to prove that any minimal fvs S of G'' that satisfies $V(G) \setminus S = F$ has a probability of being "too large" (that is, violating our claimed bound) much smaller than $2^{-|V(G)|}$. If we achieve this, then we can take a union bound over all sets F and conclude that with high probability no minimal fvs of G'' is too large.

Suppose then that we have fixed an acyclic set $F \subseteq V(G)$. We have $|F| \leq 2\alpha(G) \leq 2n^{1+\epsilon}r^{2\epsilon}$. Any minimal fvs with $V(G) \setminus S = F$ can only contain vertices from a set I_{uv} if $u, v \in F$. The total number of such vertices in G' is at most $O(n^{2+2\epsilon}r^{1/2+4\epsilon})$. The expected number of such vertices

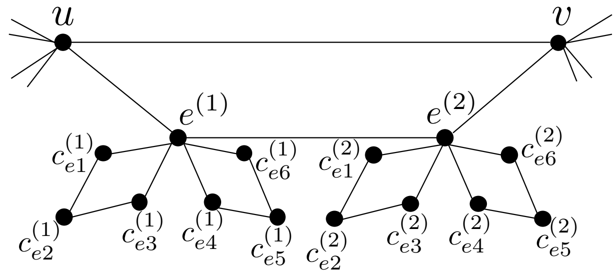

Figure 2: The edge gadget of $e = (u, v)$ in the constructed graph G.

that survive in G'' is (for some constant c) at most $\mu = cn^{1+2\epsilon}r^{1/2+4\epsilon}$. Now, using the Chernoff bound cited above we have $Pr[|X - \mu| \geq \frac{\mu}{2}] \leq 2e^{-\mu/12}$. We claim $2e^{-\mu/12} = o(2^{-|V(G)|})$. Indeed, this follows because $|V(G)| = n^{1+\epsilon}r^{1/2+\epsilon} = o(\mu)$. As a result, the probability that a large minimal fvs exists for a fixed set $F \subseteq V(G)$ exists is low enough that taking the union bound over all possible sets F we have that with high probability no minimal fvs exists with value higher than $3\mu/2$, which establishes the third property. □

5.3. NP-hardness for $\Delta = 6$

Theorem 7. Max Min FVS *is NP-hard on planar bipartite graphs with* $\Delta = 6$.

Proof. We give a reduction from Max Min VC, which is NP-hard on planar bipartite graphs of maximum degree 3 [43]. Note that the NP-hardness in [43] is stated for Minimum Independent Dominating Set, but any independent dominating set is also a maximal independent set (and vice-versa) and the complement of the minimum maximal independent set of any graph is a maximum minimal vertex cover. Thus, we also obtain NP-hardness for Max Min VC on the same instances.

We are given a graph $G = (V, E)$. For each edge $e = (u, v) \in E$, we add a path of length three from u to v going through two new vertices $e^{(1)}, e^{(2)}$ (see Figure 2). Note that $u, e^{(1)}, e^{(2)}, v$ form a cycle of length 4. Then we add two cycles of length 4, $e^{(i)}, c^{(i)}_{e1}, c^{(i)}_{e2}, c^{(i)}_{e3}$ and $e^{(i)}, c^{(i)}_{e4}, c^{(i)}_{e5}, c^{(i)}_{e6}$ for $i \in \{1, 2\}$. Let $G' = (V', E')$ be the constructed graph. Because $\Delta(G) = 3$, we have $\Delta(G') = 6$. Moreover, since G is planar and bipartite, G' is also planar and bipartite. We will show that there is a minimal vertex cover of size at least k in G if and only if there is a minimal feedback vertex set of size at least $k + 4|E|$ in G'.

Given a minimal vertex cover S of size at least k in G, we construct the set $S' = S \cup \bigcup_{e\in E}\{c_{e1}^{(1)}, c_{e4}^{(1)}, c_{e1}^{(2)}, c_{e4}^{(2)}\}$. Then $|S'| \geq k + 4|E|$. Let us first argue that S' is an fvs of G'. For each $e = (u, v) \in E$ we have at least one of $u, v \in S$, without loss of generality let $u \in S$. Now in $G'[V' \setminus S']$ the edges $(e^{(1)}, e^{(2)})$ and $(e^{(2)}, v)$ are bridges and therefore cannot be part of any cycle. The remaining cycles going through $e^{(1)}, e^{(2)}$ are handled by $\{c_{e1}^{(1)}, c_{e4}^{(1)}, c_{e1}^{(2)}, c_{e4}^{(2)}\}$. Furthermore, since $G'[V \setminus S]$ is an independent set, it is also acyclic. To see that S' is a minimal fvs, we remark that for each $c_{e1}^{(i)}, c_{e4}^{(i)}$ contained in S' there is a private cycle in $G'[V' \setminus S']$. We also note that since S is a minimal vertex cover of G, for each $u \in S$, there exists $v \notin S$ with $e = (u, v) \in E$. This means that u has the private cycle formed by $\{u, v, e^{(1)}, e^{(2)}\}$ in $G'[V' \setminus S']$. Therefore, S' is a minimal fvs.

Conversely, suppose we are given a minimal fvs S' of G' with $|S'| \geq k + 4|E|$. We will edit S' so that is contains only vertices in $V' \setminus \bigcup_{e\in E}\{e^{(1)}, e^{(2)}\}$, without decreasing its size.

First, suppose $e^{(1)}, e^{(2)} \in S'$, for some $e \in E$. We construct a new minimal fvs $S'' = S' \setminus \{e^{(2)}\} \cup \{c_{e1}^{(2)}, c_{e4}^{(2)}\}$ which is larger that S', since by minimality we have $c_{ei}^{(2)} \notin S'$ for $i \in \{1, \dots, 6\}$. It is not hard to see that S'' is indeed an fvs, as no cycle can go through $e^{(2)}$ in $G'[V' \setminus S'']$. The two vertices we added have a private cycle, while all vertices of $S' \cap S''$ retain their private cycles, so S'' is a minimal fvs. As a result in the remainder we assume that S' contains at most one of $\{e^{(1)}, e^{(2)}\}$ for all $e \in E$.

Suppose now that for some $e = (u, v) \in E$, we have $S' \cap \{u, v\} \neq \emptyset$ and $S' \cap \{e^{(1)}, e^{(2)}\} \neq \emptyset$. Without loss of generality, let $e^{(1)} \in S'$. We set $S'' = S' \setminus \{e^{(1)}\} \cup \{c_{e1}^{(1)}, c_{e4}^{(1)}\}$ and claim that S'' is a larger minimal fvs than S. Indeed, no cycle goes through $e^{(1)}$ in $G'[V' \setminus S'']$, the new vertices we added to S' have private cycles, and all vertices of $S' \cap S''$ retain their private cycles in $G'[V' \setminus S'']$. Therefore, we can now assume that if for some $e = (u, v) \in E$ we have $S' \cap \{e^{(1)}, e^{(2)}\} \neq \emptyset$ then $u, v \notin S'$.

For the remaining case, suppose that for some $e = (u, v) \in E$ we have $u, v \notin S'$ and (without loss of generality) $e^{(1)} \in S'$. We construct the set $S'' = S' \setminus \{e^{(1)}\} \cup \{c_{e1}^{(1)}, c_{e4}^{(2)}, u\}$. Note that $|S''| \geq |S'| + 2$. It is not hard to see that S'' is an fvs, since by adding $c_{e1}^{(1)}, c_{e4}^{(1)}, u$ to our set we have hit all cycles containing $e^{(1)}$ in G'. The problem now is that S'' is not necessarily minimal. We greedily delete vertices from S'' to obtain a minimal fvs S^*. We claim that in this process we cannot delete more than two vertices, that is $|S^* \setminus S''| \leq 2$. To see this, we first note that $c_{e1}^{(1)}, c_{e4}^{(2)}, u$ cannot be removed from S'' as they have private cycles in $G[V' \setminus S'']$. Suppose now that

$w_1 \in S'' \setminus S^*$ is the first vertex we removed from S'', so $G'[(V' \setminus S'') \cup \{w_1\}]$ is acyclic. This vertex must have had a private cycle in $G'[V' \setminus S']$, which was necessarily going through u. Therefore, $G'[(V' \setminus S'') \cup \{w_1\}]$ has a path connecting two neighbors of u and this path does not exist in $G'[(V' \setminus S'')]$. With a similar reasoning, removing another vertex $w_2 \in S''$ from the fvs will create a second path between neighbors of u in the induced forest. We conclude that this cannot happen a third time, since $|N(u)| \leq 3$, and if we create three paths between neighbors of u, this will create a cycle. As a result, $|S^*| \geq |S'|$. We assume in the remainder that S' does not contain $e^{(1)}, e^{(2)}$ for any $e \in E$.

Now, given a minimal fvs S' of G' with $|S'| \geq k + 4|E|$ and $S' \cap (\cup_{e \in E}\{e^{(1)}, e^{(2)}\}) = \emptyset$ we set $S = S' \cap V$ and claim that S is a minimal vertex cover of G with $|S| \geq k$. Indeed S is a vertex cover, as for each $e = (u, v) \in E$, if $u, v \notin S'$ then we would get the cycle formed by $\{u, v, e^{(1)}, e^{(2)}\}$. To see that S is minimal, suppose $N_G[u] \subseteq S'$. We claim that in that case u has no private cycle in $G'[V' \setminus S']$ (this can be seen by deleting all bridges in $G'[V' \setminus S']$, which leaves u isolated). This contradicts the minimality of S'. Finally, we argue that $|S' \setminus V| \leq 4|E|$, which gives the desired bound on $|S|$. Consider an $e = (u, v) \in E$. S' cannot contain more than one vertex among $c_{e1}^{(1)}, c_{e2}^{(1)}, c_{e3}^{(1)}$, since any of these vertices hits the cycle that goes through the others. With similar reasoning for the three other length-four cycles we conclude that S' contains at most 4 vertices for each edge $e \in E$. $\square$

6. Conclusions

We have essentially settled the approximability of Max Min FVS for polynomial and sub-exponential time, up to sub-polynomial factors in the exponent of the running time. It would be interesting to see if the running time of our sub-exponential approximation algorithm can be improved by poly-logarithmic factors in the exponent, as in [4]. In particular, improving the running time to $2^{O(n/r^{3/2})}$ seems feasible, but would likely require a version of Lemma 8 which uses more sophisticated techniques, such as Cut&Count [7, 15, 17]. For the parameterized complexity perspective, we gave a cubic kernel when parameterized by solution size. A natural direction of future work is the deep analysis of parameterized complexity of Max Min FVS. Finally, we showed that Max Min FVS is NP-hard even on graphs of maximum degree 6. An interesting open problem is the complexity on graphs of maximum degree 3, where Min FVS can be solved in polynomial time [40].

Another problem of similar spirit which deserves to be studied is Max Min OCT, where an odd cycle transversal (OCT) is a set of vertices whose removal makes the graph bipartite. This problem could also potentially be "between" Max Min VC and UDS, but obtaining a $n^{1-\epsilon}$ approximation for it seems much more challenging than for Max Min FVS.

Acknowledgments

This work is partially supported by PRC CNRS JSPS project PARAGA and by JSPS KAKENHI Grant Number JP19K21537.

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.

[2] Pierre Aboulker, Édouard Bonnet, Eun Jung Kim, and Florian Sikora. Grundy coloring & friends, half-graphs, bicliques. In *STACS*, volume 154 of *LIPIcs*, pages 58:1–58:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[3] Esther M. Arkin, Michael A. Bender, Joseph S. B. Mitchell, and Steven Skiena. The lazy bureaucrat scheduling problem. *Inf. Comput.*, 184(1):129–146, 2003.

[4] Nikhil Bansal, Parinya Chalermsook, Bundit Laekhanukit, Danupon Nanongkai, and Jesper Nederlof. New tools and connections for exponential-time approximation. *Algorithmica*, 81(10):3993–4009, 2019.

[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. *Theoretical Computer Science*, 717:2–25, 2018.

[6] Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Yota Otachi. Grundy Distinguishes Treewidth from Pathwidth. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms (ESA 2020)*, volume 173 of

Leibniz International Proceedings in Informatics (LIPIcs), pages 14:1–14:19, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[7] 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.

[8] Édouard Bonnet, Michael Lampis, and Vangelis Th. Paschos. Time-approximation trade-offs for inapproximable problems. *Journal of Computer and System Sciences*, 92:171 – 180, 2018.

[9] Nicolas Boria, Federico Della Croce, and Vangelis Th. Paschos. On the max min vertex cover problem. *Discrete Applied Mathematics*, 196:62–71, 2015.

[10] Nicolas Bourgeois, Bruno Escoffier, and Vangelis Th. Paschos. Approximation of min coloring by moderately exponential algorithms. *Inf. Process. Lett.*, 109(16):950–954, 2009.

[11] Arman Boyaci and Jérôme Monnot. Weighted upper domination number. *Electron. Notes Discret. Math.*, 62:171–176, 2017.

[12] Parinya Chalermsook, Bundit Laekhanukit, and Danupon Nanongkai. Independent set, induced matching, and pricing: Connections and tight (subexponential time) approximation hardnesses. In *FOCS*, pages 370–379. IEEE Computer Society, 2013.

[13] Grant A. Cheston, Gerd Fricke, Stephen T. Hedetniemi, and David Pokrass Jacobs. On the computational complexity of upper fractional domination. *Discret. Appl. Math.*, 27(3):195–207, 1990.

[14] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

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

[16] Marek Cygan, Lukasz Kowalik, and Mateusz Wykurz. Exponential-time approximation of weighted set cover. *Inf. Process. Lett.*, 109(16):957–961, 2009.

[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. In *FOCS*, pages 150–159. IEEE Computer Society, 2011.

[18] Marek Cygan and Marcin Pilipczuk. Exact and approximate bandwidth. *Theor. Comput. Sci.*, 411(40-42):3701–3713, 2010.

[19] Marc Demange. A note on the approximation of a minimum-weight maximal independent set. *Computational Optimization and Applications*, 14(1):157–169, 1999.

[20] Louis Dublois, Tesshu Hanaka, Mehdi Khosravian Ghadikolaei, Michael Lampis, and Nikolaos Melissinos. (In)approximability of Maximum Minimal FVS. In Yixin Cao, Siu-Wing Cheng, and Minming Li, editors, *31st International Symposium on Algorithms and Computation (ISAAC 2020)*, volume 181 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[21] Louis Dublois, Michael Lampis, and Vangelis Th. Paschos. Upper dominating set: Tight algorithms for pathwidth and sub-exponential approximation. *CoRR*, abs/2101.07550, 2021.

[22] Bruno Escoffier, Vangelis Th. Paschos, and Emeric Tourniaire. Approximating MAX SAT by moderately exponential and parameterized algorithms. *Theor. Comput. Sci.*, 560:147–157, 2014.

[23] Hiroshi Eto, Tesshu Hanaka, Yasuaki Kobayashi, and Yusuke Kobayashi. Parameterized Algorithms for Maximum Cut with Connectivity Constraints. In Bart M. P. Jansen and Jan Arne Telle, editors, *14th International Symposium on Parameterized and Exact Computation (IPEC 2019)*, volume 148 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[24] Dimitris Fotakis, Michael Lampis, and Vangelis Th. Paschos. Subexponential approximation schemes for csps: From dense to almost sparse. In *STACS*, volume 47 of *LIPIcs*, pages 37:1–37:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[25] Fabio Furini, Ivana Ljubic, and Markus Sinnl. An effective dynamic programming algorithm for the minimum-cost maximal knapsack packing

problem. *European Journal of Operational Research*, 262(2):438–448, 2017.

[26] 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.

[27] Tesshu Hanaka, Hans L. Bodlaender, Tom C. van der Zanden, and Hirotaka Ono. On the maximum weight minimal separator. *Theoretical Computer Science*, 796:294 – 308, 2019.

[28] Ararat Harutyunyan, Mehdi Khosravian Ghadikolaei, Nikolaos Melissinos, Jérôme Monnot, and Aris Pagourtzis. On the complexity of the upper r-tolerant edge cover problem. In Luís Soares Barbosa and Mohammad Ali Abam, editors, *Topics in Theoretical Computer Science - Third IFIP WG 1.8 International Conference, TTCS 2020, Tehran, Iran, July 1-2, 2020, Proceedings*, volume 12281 of *Lecture Notes in Computer Science*, pages 32–47. Springer, 2020.

[29] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Math*, 182:105–142, 1999.

[30] Michael A. Henning and Dinabandhu Pradhan. Algorithmic aspects of upper paired-domination in graphs. *Theor. Comput. Sci.*, 804:98–114, 2020.

[31] Michael A. Henning and Anders Yeo. On upper transversals in 3-uniform hypergraphs. *Electron. J. Comb.*, 25(4):P4.27, 2018.

[32] Michael A. Henning and Anders Yeo. Upper transversals in hypergraphs. *Eur. J. Comb.*, 78:1–12, 2019.

[33] Michael A. Henning and Anders Yeo. Bounds on upper transversals in hypergraphs. *J. Comb. Optim.*, 39(1):77–89, 2020.

[34] Ken Iwaide and Hiroshi Nagamochi. An improved algorithm for parameterized edge dominating set problem. *J. Graph Algorithms Appl.*, 20(1):23–58, 2016.

[35] Michael S. Jacobson and Kenneth Peters. Chordal graphs and upper irredundance, upper domination and independence. *Discret. Math.*, 86(1-3):59–69, 1990.

[36] Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Improved (in-)approximability bounds for d-scattered set. In *WAOA*, volume 11926 of *Lecture Notes in Computer Science*, pages 202–216. Springer, 2019.

[37] Kaveh Khoshkhah, Mehdi Khosravian Ghadikolaei, Jérôme Monnot, and Florian Sikora. Weighted upper edge cover: Complexity and approximability. *J. Graph Algorithms Appl.*, 24(2):65–88, 2020.

[38] 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.

[39] Pranabendu Misra, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Parameterized algorithms for even cycle transversal. In Martin Charles Golumbic, Michal Stern, Avivit Levy, and Gila Morgenstern, editors, *Graph-Theoretic Concepts in Computer Science - 38th International Workshop, WG 2012, Jerusalem, Israel, June 26-28, 2012, Revised Selcted Papers*, volume 7551 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 2012.

[40] Shuichi Ueno, Yoji Kajitani, and Shin'ya Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, 72(1):355 – 360, 1988.

[41] Meirav Zehavi. Maximum minimal vertex cover parameterized by vertex cover. *SIAM Journal on Discrete Mathematics*, 31(4):2440–2456, 2017.

[42] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.

[43] Igor E. Zvervich and Vadim E. Zverovich. An induced subgraph characterization of domination perfect graphs. *Journal of Graph Theory*, 20(3):375–395, 1995.