# Simultaneous Feedback Vertex Set: A Parameterized Perspective[*]

AKANKSHA AGRAWAL, University of Bergen, Norway
DANIEL LOKSHTANOV, University of Bergen, Norway
AMER E. MOUAWAD[†], University of Bergen, Norway
SAKET SAURABH, University of Bergen, Norway and The Institute of Mathematical Sciences, India

For a family $\mathcal{F}$ of graphs, a graph $G$, and a positive integer $k$, the $\mathcal{F}$-Deletion problem asks whether we can delete at most $k$ vertices from $G$ to obtain a graph in $\mathcal{F}$. $\mathcal{F}$-Deletion generalizes many classical graph problems such as Vertex Cover, Feedback Vertex Set, and Odd Cycle Transversal. For an integer $\alpha \geq 1$, an $n$-vertex (multi) graph $G = (V, \bigcup_{i=1}^{\alpha} E_i)$, where the edge set of $G$ is partitioned into $\alpha$ color classes, is called an $\alpha$-edge-colored (multi) graph. A natural extension of the $\mathcal{F}$-Deletion problem to edge-colored graphs is the Simultaneous $\mathcal{F}$-Deletion problem. In the latter problem, we are given an $\alpha$-edge-colored graph $G$ and the goal is to find a set $S$ of at most $k$ vertices such that each graph $G_i - S$, where $G_i = (V, E_i)$ and $1 \leq i \leq \alpha$, is in $\mathcal{F}$. In this work, we study Simultaneous $\mathcal{F}$-Deletion for $\mathcal{F}$ being the family of forests. In other words, we focus on the Simultaneous Feedback Vertex Set (SimFVS) problem. Algorithmically, we show that, like its classical counterpart, SimFVS parameterized by $k$ is fixed-parameter tractable (FPT) and admits a polynomial kernel, for any fixed constant $\alpha$. In particular, we give an algorithm running in $2^{O(\alpha k)} n^{O(1)}$ time and a kernel with $O(\alpha k^{3(\alpha+1)})$ vertices. The running time of our algorithm implies that SimFVS is FPT even when $\alpha \in o(\log n)$. We complement this positive result by showing that if we allow $\alpha$ to be in $O(\log n)$, where $n$ is the number of vertices in the input graph, SimFVS becomes W[1]-hard. In particular, when $\alpha$ is roughly equal to $c \log n$, for a non-zero positive constant $c$, the problem becomes W[1]-hard. Our positive results answer one of the open problems posed by Cai and Ye (MFCS 2014).

CCS Concepts: • **Theory of computation → Graph algorithms analysis**; **Fixed parameter tractability**; *Complexity classes*; *Problems, reductions and completeness*; *Backtracking*; *Branch-and-bound*;

Additional Key Words and Phrases: parameterized complexity, feedback vertex set, kernel, edge-colored graphs

**ACM Reference Format:**
Akanksha Agrawal, Daniel Lokshtanov, Amer E. Mouawad, and Saket Saurabh. 2018. Simultaneous Feedback Vertex Set: A Parameterized Perspective. *ACM Trans. Comput. Theory* 9, 4, Article 39 (March 2018), 25 pages. https://doi.org/0000001.0000001

[*] A preliminary version of this work appeared in the proceedings of the 33rd International Symposium on Theoretical Aspects of Computer Science (STACS 2016).
[†] The corresponding author

Author's addresses: A. Agrawal, D. Lokshtanov, A. E. Mouawad, and S. Saurabh, Department of Informatics, University of Bergen, Norway; S. Saurabh, Institute of Mathematical Sciences, Chennai, India.
Authors' addresses: Akanksha Agrawal, akanksha.agrawal@uib.no, University of Bergen, Bergen, Norway; Daniel Lokshtanov, daniel.lokshtanov@uib.no, University of Bergen, Bergen, Norway; Amer E. Mouawad, a.mouawad@uib.no, University of Bergen, Bergen, Norway; Saket Saurabh, saket@imsc.res.in, University of Bergen, Bergen, Norway, The Institute of Mathematical Sciences, HBNI, Chennai, India.

1942-3454/2018/3-ART39 $15.00
https://doi.org/0000001.0000001

## 1 INTRODUCTION

In graph theory, one can define a general family of problems as follows. Let $\mathcal{F}$ be a collection of graphs. Given an undirected graph $G$ and a positive integer $k$, is it possible to perform at most $k$ edit operations to $G$ so that the resulting graph belongs to $\mathcal{F}$? Here one can define edit operations as either vertex/edge deletions, edge additions, or edge contractions. Such problems constitute a large fraction of problems considered under the parameterized complexity framework and have been studied extensively. When edit operations are restricted to vertex deletions this corresponds to the $\mathcal{F}$-Deletion problem, which generalizes classical graph problems such as Vertex Cover [6], Feedback Vertex Set [5, 8, 18], Vertex Planarization [24], Odd Cycle Transversal [19, 21], Interval Vertex Deletion [4], Chordal Vertex Deletion [23], and Planar $\mathcal{F}$-Deletion [11, 17]. The topic of this paper is a generalization of $\mathcal{F}$-Deletion problems to "edge-colored graphs". In particular, we do a case study of an edge-colored version of the classical Feedback Vertex Set problem [12].

A graph $G = (V, \bigcup_{i=1}^{\alpha} E_i)$, where the edge set of $G$ is partitioned into $\alpha$ color classes, is called an *$\alpha$-edge-colored graph*. As stated by Cai and Ye [3], "edge-colored graphs are fundamental in graph theory and have been extensively studied in the literature, especially for alternating cycles, monochromatic subgraphs, heterochromatic subgraphs, and partitions". A natural extension of the $\mathcal{F}$-Deletion problem to edge-colored graphs is the Simultaneous $\mathcal{F}$-Deletion problem. In the latter problem, we are given an $\alpha$-edge-colored graph $G$ and the goal is to find a set $S$ of at most $k$ vertices such that each graph $G_i - S$, where $G_i = (V, E_i)$ and $1 \leq i \leq \alpha$, is in $\mathcal{F}$. Cai and Ye [3] studied several problems restricted to 2-edge-colored graphs, where edges are colored either red or blue. In particular, they consider the Dually Connected Induced Subgraph problem, i.e. find a set $S$ of $k$ vertices in $G$ such that both induced graphs $G_{\text{red}}[S]$ and $G_{\text{blue}}[S]$ are connected, and the Dual Separator problem, i.e. delete a set $S$ of at most $k$ vertices to simultaneously disconnect the red and blue graphs of $G$. They show, among other results, that Dual Separator is NP-complete and Dually Connected Induced Subgraph is W[1]-hard even when both $G_{\text{red}}$ and $G_{\text{blue}}$ are trees. One of the open problems they state is to determine the parameterized complexity of Simultaneous $\mathcal{F}$-Deletion for $\alpha = 2$ and $\mathcal{F}$ the family of forests, bipartite graphs, chordal graphs, or planar graphs. The focus in this work is on one of those problems, namely Simultaneous Feedback Vertex Set, an interesting, and well-motivated [2, 3, 16], generalization of Feedback Vertex Set on edge-colored graphs.

A *feedback vertex set* is a subset $S$ of vertices such that $G - S$ is a forest. For an $\alpha$-edge-colored graph $G$, an *$\alpha$-simultaneous feedback vertex set* (or *$\alpha$-simfvs* for short) is a subset $S$ of vertices such that $G_i - S$ is a forest for each $1 \leq i \leq \alpha$. The Simultaneous Feedback Vertex Set is stated formally as follows.

Simultaneous Feedback Vertex Set (SimFVS) **Parameter:** $k$
**Input:** $(G = (V, \bigcup_{i=1}^{\alpha} E_i), k)$, where $G$ is an undirected $\alpha$-edge-colored (multi) graph and $k$ is a positive integer.
**Question:** Is there a subset $S \subseteq V(G)$ of size at most $k$ such that for $1 \leq i \leq \alpha$, $G_i - S$ is a forest?

Given a graph $G = (V, E)$ and a positive integer $k$, the classical Feedback Vertex Set (FVS) problem asks whether there exists a set $S$ of at most $k$ vertices in $G$ such that the graph induced on $V(G) \setminus S$ is acyclic. In other words, the goal is to find a set of at most $k$ vertices that intersects all cycles in $G$. FVS is a classical NP-complete [12] problem with numerous applications and is by now very well understood from both the classical and parameterized complexity [10] view points. For instance, the problem admits a 2-approximation algorithm [1], an exact (non-parameterized) algorithm running in $O(1.736^n n^{O(1)})$ time [29], a deterministic algorithm running in $O^\star(3.619^k)$ time [18][1], a randomized algorithm running in $O^\star(3^k)$ time [8], and a kernel on $O(k^2)$ vertices [27] (see Section 2 for definitions).

**Our results and methods.** We show that, like its classical counterpart, SimFVS parameterized by $k$ is FPT and admits a polynomial kernel, for any fixed constant $\alpha$. In particular, we obtain the following results.

- An FPT algorithm running in $O^\star(23^{\alpha k})$ time. For the special case of $\alpha = 2$, we give a faster algorithm running in $O^\star(81^k)$ time. Using a completely different approach, Ye [30] independently (and simultaneously) gave an algorithm running in $O^\star(k^{\alpha k})$ time for the general case and an algorithm running in $O^\star(c^k)$ time, $c > 81$, for the case $\alpha = 2$.
- For constant $\alpha$, we obtain a kernel with $O(\alpha k^{3(\alpha+1)})$ vertices.
- The running time of our algorithm implies that SimFVS is FPT even when $\alpha \in o(\log n)$. We complement this positive result by showing that for $\alpha \in O(\log n)$, where $n$ is the number of vertices in the input graph, SimFVS becomes W[1]-hard. In particular, when $\alpha$ is roughly equal to $c \log n$, for a non-zero positive constant $c$, the problem becomes W[1]-hard.

Our algorithms and kernel build on the tools and methods developed for FVS [7]. However, we need to develop both new branching rules as well as new reduction rules. The main reason why our results do not follow directly from earlier work on FVS is the following. Many (if not all) parameterized algorithms, as well as kernelization algorithms, developed for the FVS problem [7] exploit the fact that vertices of degree two or less in the input graph are, in some sense, irrelevant. In other words, vertices of degree one or zero cannot participate in any cycle and every cycle containing any degree-two vertex must contain both of its neighbors. Hence, if this degree-two vertex is part of a feedback vertex set then it can be replaced by either one of its neighbors. Unfortunately (or fortunately for us), this property does not hold for the SimFVS problem when we restrict our attention to edges of the same color, even on graphs where edges are bicolored either red or blue. For instance, if a vertex is incident to two red edges and two blue edges, it might in fact be participating in two distinct cycles. Hence, it is not possible to neglect (or shortcut) this vertex in neither $G_{\text{red}}$ nor $G_{\text{blue}}$. As we shall see, most of the new algorithmic techniques that we present deal with vertices of exactly this type. Although very tightly related to one another, we show that there are subtle and interesting differences separating the FVS problem from the SimFVS problem, even for $\alpha = 2$. For this reason, we also believe that studying Simultaneous $\mathcal{F}$-Deletion for different families of graphs $\mathcal{F}$, e.g. bipartite, chordal, or planar graphs, might reveal some new insights about the classical underlying problems.

In Section 3, we present an algorithm solving the SimFVS problem, parameterized by solution size $k$, in $O^\star(23^{\alpha k})$ time. Our algorithm follows the iterative compression paradigm introduced by Reed et al. [26] combined with new reduction and branching rules. Our main new branching rule can be described as follows: Given a maximal degree-two path in some $G_i$, $1 \leq i \leq \alpha$, we branch depending on whether there is a vertex from this path participating in an $\alpha$-simultaneous feedback vertex set or not. In the branch where we guess that a solution contains a vertex from this path, we construct a color $i$ cycle which is isolated from the rest of the graph. In the other branch, we are able to follow known strategies by "simulating" the classical FVS problem. Observe that we can never have more than $k$ isolated cycles of the same color in a positive instance. Hence, by incorporating this fact into our measure we are guaranteed to make "progress" in both branches. For the base case, each $G_i$ is a disjoint union of cycles (though not $G$) and to find an $\alpha$-simultaneous feedback

[1] We use the $O^\star$ notation to describe the running times of our algorithms. A running time $O^\star(f(k))$ means that the running time is upper bounded by $f(k)n^{O(1)}$, where $n$ is the input size. That is, the $O^\star$ notation suppresses factors that are polynomial in the input size from the running-time expression.

vertex set for $G$ we cast the remaining problem as an instance of Hitting Set parameterized by the size of the family. For $\alpha = 2$, we can instead use an algorithm for finding maximum matchings in an auxiliary graph. Using this fact we give a faster, $O^\star(81^k)$ time, algorithm for the case $\alpha = 2$.

In Section 4, we tackle the question of kernelization and present a polynomial kernel for the problem, for constant $\alpha$. Our kernel has $O(\alpha k^{3(\alpha+1)})$ vertices and requires new insights into the possible structures induced by those special vertices discussed above. In particular, we enumerate all maximal degree-two paths in each $G_i$ after deleting an approximate feedback vertex set in $G_i$ and study how such paths interact with each other. Using vertex "marking" techniques, we are able to "unwind" long degree-two paths by making a private copy of each unmarked vertices for each color class. This unwinding leads to "normal" degree-two paths on which classical reduction rules can be applied and hence we obtain the desired kernel.

Finally, we consider the dependence between $\alpha$ and both the size of our kernel and the running time of our algorithm in Section 5. We show that even for $\alpha \in O(\log n)$, where $n$ is the number of vertices in the input graph, SimFVS becomes W[1]-hard. We show hardness via a new problem of independent interest which we denote by Partitioned Hitting Set. Partitioned Hitting Set is a special variant of the well-known Hitting Set problem. In the Hitting Set problem, we are given a universe $\mathcal{U}$, a family $\mathcal{F} = \{f_1, f_2, \ldots\}$ of subsets of $\mathcal{U}$, and an integer $k$, and the goal is to "hit" every set in $\mathcal{F}$ using at most $k$ elements from $\mathcal{U}$. Formally, the goal is to determine whether there exists $U \subseteq \mathcal{U}$ such that $|U| \leq k$ and $U \cap f \neq \emptyset$, for all $f \in \mathcal{F}$. The input to the Partitioned Hitting Set problem consists of a tuple $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k)$, where each $\mathcal{F}_i$, $1 \leq i \leq \alpha$, is a collection of subsets of the finite universe $\mathcal{U}$, $k$ is a positive integer, and all the sets within a family $\mathcal{F}_i$, $1 \leq i \leq \alpha$, are pairwise disjoint. As in the Hitting Set problem, the goal is to determine whether there exists a subset $U$ of $\mathcal{U}$ of cardinality at most $k$ such that for every $f \in \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha$, $f \cap U$ is nonempty. We show that Partitioned Hitting Set is W[1]-hard for $\alpha \in O(\log |\mathcal{U}||\mathcal{F}|)$ via a reduction from Partitioned Subgraph Isomorphism and we show that SimFVS is W[1]-hard for $\alpha \in O(\log n)$ via a reduction from Partitioned Hitting Set with $\alpha \in O(\log |\mathcal{U}||\mathcal{F}|)$. Along the way, we also show, using a somewhat simpler reduction from Hitting Set, that SimFVS is W[2]-hard for $\alpha \in O(n)$.

## 2 PRELIMINARIES

We start with some basic definitions and introduce terminology from graph theory and algorithms. We also establish some of the notation that will be used throughout.

**Graphs.** For a graph $G$, by $V(G)$ and $E(G)$ we denote its vertex set and edge set, respectively. We only consider finite graphs possibly having loops and multiedges. In the following, let $G$ be a graph and let $H$ be a subgraph of $G$. By $d_H(v)$, we denote the degree of vertex $v$ in $H$. For any non-empty subset $W \subseteq V(G)$, the subgraph of $G$ induced by $W$ is denoted by $G[W]$; its vertex set is $W$ and its edge set consists of all those edges of $E$ with both endpoints in $W$. For $W \subseteq V(G)$, by $G - W$ we denote the graph obtained by deleting the vertices in $W$ and all edges which are incident to at least one vertex in $W$.

A *path* in a graph is a sequence of distinct vertices $v_0, v_1, \ldots, v_k$ such that $(v_i, v_{i+1})$ is an edge for all $0 \leq i < k$. A *cycle* in a graph is a sequence of distinct vertices $v_0, v_1, \ldots, v_k$ such that $(v_i, v_{i+1})$ is an edge for all $0 \leq i < k$ and $(v_k, v_0)$ is an edge. We note that both a double edge and a loop are cycles. We also use the convention that a loop at a vertex $v$ contributes two to the degree of $v$.

An $\alpha$-edge-colored graph (or $\alpha$-colored graph for short) is a graph $G = (V, \bigcup_{i=1}^{\alpha} E_i)$. We call $G_i$ the color $i$ (or $i$-color) graph of $G$, where $G_i = (V, E_i)$ (the graph $G$ is the "bag" union of all graphs $G_i$). For notational convenience we sometimes denote an $\alpha$-colored graph by $G = (V, E_1, E_2, ..., E_\alpha)$. For an $\alpha$-colored graph $G$, the *total degree* of a vertex $v$ is $\sum_{i=1}^{\alpha} d_{G_i}(v)$. A vertex $v$ is *isolated* if it has total degree zero (a vertex with a loop is not isolated). By color $i$ edge (or $i$-color edge) we refer to an edge in $E_i$, for $1 \leq i \leq \alpha$. A vertex $v \in V(G)$ is said to have a color $i$ neighbor if there is an edge $(v, u)$ in $E_i$, furthermore $u$ is a color $i$ neighbor of $v$. We say a path or a cycle in $G$ is *monochromatic* if all the edges on the path or cycle have the same color. Given a vertex $v \in V(G)$, a *$v$-flower* of order $k$ is a set of $k$ cycles in $G$ whose pairwise intersection is exactly $\{v\}$. If all cycles in a $v$-flower are monochromatic then we have a *monochromatic $v$-flower.* An $\alpha$-colored graph $G = (V, E_1, E_2, \cdots, E_\alpha)$ is an *$\alpha$-forest* if each $G_i$ is a forest, for $1 \leq i \leq \alpha$. We refer the reader to [9] for details on standard graph theoretic notation and terminology we use in the paper.

**Parameterized complexity.** A *parameterized problem* $\Pi$ is a subset of $\Gamma^* \times \mathbb{N}$, where $\Gamma$ is a finite alphabet. An instance of a parameterized problem is a tuple $(x, k)$, where $x$ is a classical problem instance, and $k$ is called the *parameter.* A central notion in parameterized complexity is *fixed-parameter tractability (FPT)* which means, for a given instance $(x, k)$, decidability in time $f(k) \cdot p(|x|, k)$, where $f$ is a computable function of $k$ and $p$ is a polynomial in $|x|$ and $k$. For showing that a given parameterized problem is not likely to be FPT, i.e. not likely to admit an algorithm running in $f(k) \cdot p(|x|, k)$ time, Downey and Fellows [10] introduced the W-hierarchy. The hierarchy consists of a complexity class W$[t]$, for every integer $t \geq 1$, such that W$[t] \subseteq$ W$[t+1]$ for all $t$. Downey and Fellows [52] proved that FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ W$[t]$ and conjectured that strict containment holds. In particular, the assumption FPT $\subset$ W[1] is a natural parameterized analogue of the conjecture that P $\neq$ NP. Similarly to the notion of reductions in classic complexity theory, we can use the notion of parameterized reductions for showing that a problem is FPT or for placing it inside the W-hierarchy. Let $A, B \subseteq \Gamma^* \times \mathbb{N}$ be two parameterized problems. A *parameterized reduction* from $A$ to $B$ is an algorithm that, given an instance $(x, k)$ of $A$, outputs an instance $(x', k')$ of $B$ such that $(x, k)$ is a yes-instance of $A$ if and only if $(x', k')$ is a yes-instance of $B$, $k' \leq g(k)$ for some computable function $g$, and the running time is $f(k) \cdot p(|x|, k)$, where $f$ is a computable function of $k$ and $p$ is a polynomial in $|x|$ and $k$.

**Kernelization.** A kernelization algorithm for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}$ is an algorithm that, given $(x, k) \in \Gamma^* \times \mathbb{N}$, outputs, in time polynomial in $|x|$ and $k$, a pair $(x', k') \in \Gamma^* \times \mathbb{N}$ such that $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$ and $|x'|, k' \leq g(k)$, where $g$ is some computable function. The output instance $x'$ is called the kernel, and the function $g$ is referred to as the size of the kernel. If $g(k) = k^{O(1)}$ (resp. $g(k) = O(k)$) then we say that $\Pi$ admits a polynomial (resp. linear) kernel. A *reduction rule* for a parameterized problem $\Pi$ is a function $\rho : \Gamma^* \times \mathbb{N} \rightarrow \Gamma^* \times \mathbb{N}$ that maps an instance $(x, k)$ (of $\Pi$) to an instance $(x', k')$ (of $\Pi$) such that $\rho$ is computable in time polynomial in $|x|$ and $k$. We say that two instances of $\Pi$, $(x, k)$ and $(x', k')$, are *equivalent* if $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$. A reduction rule is said to be *safe* whenever it translates an instance into an equivalent one.

## 3 FPT ALGORITHM FOR SIMULTANEOUS FEEDBACK VERTEX SET

We give an algorithm for the SimFVS problem using the method of iterative compression [7, 26]. We briefly describe the general scheme of iterative compression and then focus on an algorithm for solving the disjoint version of SimFVS. We refer the reader to [7, 26] for more details on iterative compression. In the Disjoint SimFVS problem, we are given an $\alpha$-colored graph $G = (V, \bigcup_{i=1}^{\alpha} E_i)$, an integer $k$, and an $\alpha$-simfvs $W$ in $G$ of size $k+1$. The objective is to find an $\alpha$-simfvs $S \subseteq V(G) \setminus W$ of size at most $k$, or correctly conclude the non-existence of such an $\alpha$-simfvs.

Let $(G = (V, \bigcup_{i=1}^{\alpha} E_i), k)$ be an instance of SimFVS. We fix an arbitrary ordering $(v_1, v_2, \ldots, v_n)$ on the vertices of $G$. For $j \in \{1, \ldots, n\}$, we let $G^{(j)}$ denote the subgraph of $G$ induced on the first $j$ vertices. Note that when $j = k$, we can take the vertex set of $G^{(k)}$ as an $\alpha$-simfvs in $G^{(k)}$ of size

$k$. Now suppose that for some $j > k$, we have constructed an $\alpha$-simfvs $S^{(j)}$ of $G^{(j)}$ of size at most $k$. Then, in the graph $G^{(j+1)}$, the set $Z^{(j+1)} = S^{(j)} \cup \{v_{j+1}\}$ is an $\alpha$-simfvs of size at most $k + 1$. If in fact $|Z^{(j+1)}| \leq k$ then we are done, i.e. we let $S^{(j+1)} = Z^{(j+1)}$ and proceed to the next iteration. Otherwise, $|Z^{(j+1)}| = k + 1$ and we need to "compress it" into a smaller solution (if it exists). To that end, we first guess the intersection $X$ of $S^{(j+1)}$ with $Z^{(j+1)}$. In other words, for every $q \in \{0, \ldots, k\}$ and every subset $X$ of $Z^{(j+1)}$ of size $q$, we construct an instance $(G', W', k')$ of Disjoint SimFVS as follows. We let $G' = G^{(j+1)} - X$, $W' = Z^{(j+1)} \setminus X$, and $k' = k - q$. Note that $|W'| = k - q + 1$, so $|W'|$ is one larger than $k'$. If $G'$ does not admit an $\alpha$-simfvs of size at most $k$, then of course neither does $G$, and we may terminate (and declare a no-instance). If $S^{(j+1)}$ has been successfully found (which is the union of $X$ and the solution to instance $(G', W', k')$), then we proceed to the next graph $G^{(j+2)}$, and repeat. Finally, observe that $G^{(n)} = G$, so we eventually either find an $\alpha$-simfvs of size at most $k$ in $G$ or conclude that no solution exists. A simple calculation shows that the existence of an algorithm running in $c^k \cdot n^{O(1)}$ time for the disjoint variant implies that SimFVS can be solved in time $(1 + c)^k \cdot n^{O(1)}$ [7]. In Section 3.1 we describe such an algorithm for Disjoint SimFVS.

### 3.1 Algorithm for Disjoint SimFVS

Let $(G = (V, E_1, E_2, \ldots, E_\alpha), W, k)$ be an instance of Disjoint SimFVS and let $F = G - W$. We start with some simple reduction rules that clean up the graph. Whenever more than one reduction rule applies, we apply the lowest-numbered applicable rule.

- Reduction SimFVS.R1. Delete isolated vertices.
- Reduction SimFVS.R2. If there is a vertex $v$ which has only one neighbor $u$ in $G_i$, for some $i \in \{1, 2, \ldots, \alpha\}$, then delete the edge $(v, u)$ from $E_i$.
- Reduction SimFVS.R3. If there is a vertex $v \in V(G)$ whose total degree is exactly two and whose neighbors are $u$ and $w$, delete edges $(v, u)$ and $(v, w)$ from $E_i$ and add an edge $(u, w)$ to $E_i$, where $i$ is the color of edges $(v, u)$ and $(v, w)$. Note that after reduction SimFVS.R2 has been applied, both edges $(v, u)$ and $(v, w)$ must be of the same color.
- Reduction SimFVS.R4. If for some $i \in \{1, 2, \ldots, \alpha\}$ there is an edge of multiplicity larger than two in $E_i$, reduce its multiplicity to two.
- Reduction SimFVS.R5. If there is a vertex $v$ with a loop, then add $v$ to the solution set, delete $v$ (and all edges incident on $v$) from the graph and decrease $k$ by 1.

Lemma 3.1. *Reduction rule* SimFVS.*R1 is safe.*

Proof. Let $G$ be an $\alpha$-colored graph and let $v$ be an isolated vertex in G. Consider the $\alpha$-colored graph $G'$ with vertex set $V(G') = V(G) \setminus \{v\}$ and edge set $E(G') = E(G)$. Since $v$ does not participate in any cycle, it follows that $G$ has an $\alpha$-simfvs of size at most $k$ if and only if $G'$ has an $\alpha$-simfvs of size at most $k$. □

Lemma 3.2. *Reduction rule* SimFVS.*R2 is safe.*

Proof. Let $G$ be an $\alpha$-colored graph and $v$ be a vertex whose only neighbor in $G_i$ is $u$, for some $i \in \{1, 2, \ldots, \alpha\}$. Consider the $\alpha$-colored graph $G'$ with vertex set $V(G)$ and edge sets $E_i(G') = E_i(G) \setminus \{(v, u)\}$ and $E_j(G') = E_j(G)$, for $j \in \{1, 2, \ldots, \alpha\} \setminus \{i\}$. We show that $G$ has an $\alpha$-simfvs of size at most $k$ if and only if $G'$ has an $\alpha$-simfvs of size at most $k$.

In the forward direction, consider an $\alpha$-simfvs $S$ in $G$ of size at most $k$. Since $G'_j = G_j$, $S$ intersects all the cycles in $G'_j$, $j \in \{1, 2, \ldots, \alpha\} \setminus \{i\}$. Note that in $G_i$, there is no cycle containing the edge $(u, v)$ as $v$ is a degree-one vertex in $G_i$. Hence, all the cycles in $G_i$ are also cycles in $G'_i$. $S$ intersects all cycles in $G_i$ and, in particular, $S$ intersects all cycles in $G'_i$. Therefore, $S$ is an $\alpha$-simfvs in $G'$ of size at most $k$.

For the reverse direction, consider an $\alpha$-simfvs $S$ in $G'$ of size at most $k$. If $S$ is not an $\alpha$-simfvs of $G$ then there is a cycle $C$ in some $G_t$, for $t \in \{1, 2, \ldots, \alpha\}$. Note that $C$ cannot be a cycle in $G_j$ as $G_j = G'_j$, for $j \in \{1, 2, \ldots, \alpha\} \setminus \{i\}$. Therefore $C$ must be a cycle in $G_i$. The cycle $C$ must contain the edge $(v, u)$, as this is the only edge in $G_i$ which is not an edge in $G'_i$. But $v$ is a degree-one vertex in $G_i$, so it cannot be part of any cycle in $G_i$, contradicting the existence of cycle $C$. Thus $S$ is an $\alpha$-simfvs of $G$ of size at most $k$. □

Lemma 3.3. *Reduction rule* SimFVS*.R3 is safe.*

Proof. Consider an $\alpha$-colored graph $G$. Let $v$ be a vertex in $V(G)$ such that $v$ has total degree 2 and let $u, w$ be the neighbors of $v$ in $G_i$, where $u \neq w$ and $i \in \{1, 2, \ldots, \alpha\}$. Recall that after reduction SimFVS.R2 has been applied, both edges $(v, u)$ and $(v, w)$ must be of the same color. Consider the $\alpha$-colored graph $G'$ with vertex set $V(G)$ and edge sets $E_i(G') = (E_i(G) \setminus \{(v, u), (v, w)\}) \cup \{(u, w)\}$ and $E_j(G') = E_j(G)$, for $j \in \{1, 2, \ldots, \alpha\} \setminus \{i\}$. We show that $G$ has an $\alpha$-simfvs of size at most $k$ if and only if $G'$ has an $\alpha$-simfvs of size at most $k$.

In the forward direction, let $S$ be an $\alpha$-simfvs in $G$ of size at most $k$. Suppose $S$ is not an $\alpha$-simfvs of $G'$. Then, there is a cycle $C$ in $G'_t$, for some $t \in \{1, 2, \ldots, \alpha\}$. Note that $C$ cannot be a cycle in $G'_j$ as $G'_j = G_j$, for $j \in \{1, 2, \ldots, \alpha\} \setminus \{i\}$. Therefore $C$ must be a cycle in $G'_i$. All the cycles $C'$ not containing the edge $(u, w)$ are also cycles in $G_i$ and therefore $S$ must contain some vertex from $C'$. It follows that $C$ must contain the edge $(u, w)$. Note that the edges $(E(C) \setminus \{(u, w)\}) \cup \{(v, u), (w, v)\}$ form a cycle in $G_i$. Therefore $S$ must contain a vertex from $V(C) \cup \{v\}$. We consider the following cases:

- Case 1: $v \notin S$. In this case $S$ must contains a vertex from $V(C)$. Hence, $S$ is an $\alpha$-simfvs in $G'$.
- Case 2: $v \in S$. Let $S' = (S \setminus \{v\}) \cup \{u\}$. Any cycle $C'$ containing $v$ in $G_i$ must contain $u$ and $w$ (since $d_{G_i}(v) = 2$ for Reduction rule SimFVS.R3 to be applicable). But $S'$ intersects all such cycles $C'$, as $u \in S'$. Therefore $S'$ is an $\alpha$-simfvs of $G'$ of size at most $k$.

In the reverse direction, consider an $\alpha$-simfvs $S$ of $G'$. $S$ intersects all cycles in $G_j$, since $G_j = G'_j$, for $j \in \{1, 2, \ldots, \alpha\} \setminus \{i\}$. All cycles in $G_i$ not containing $v$ are also cycles in $G'_i$ and therefore $S$ intersects all such cycles. A cycle $C$ in $G_i$ containing $v$ must contain $u$ and $w$ ($v$ is a degree-two vertex in $G_i$). Note that $(E(C) \setminus \{(v, u), (v, w)\}) \cup \{(u, w)\}$ is a cycle in $G'_i$ and $S$, being an $\alpha$-simfvs in $G'$, must contain a vertex from $V(C) \setminus \{v\}$. Therefore $S \cap V(C) \neq \emptyset$, so $S$ intersects cycle $C$ in $G'_i$. Hence $S$ an $\alpha$-simfvs in $G'$. □

The safeness of reduction rule SimFVS.R4 follows from the fact that edges of multiplicity greater than two do not influence the set of feasible solutions. Safeness of reduction rule SimFVS.R5 follows from the fact that any vertex with a loop must be present in every solution (if a solution exists). Note that all of the above reduction rules can be applied in polynomial time. Moreover, after exhaustively applying all rules, the resulting graph $G$ satisfies the following properties:

(P1) $G$ contains no loops;
(P2) Every edge in $G_i$, for $i \in \{1, 2, \ldots, \alpha\}$, is of multiplicity at most two;
(P3) Every vertex in $G$ has either degree zero or degree at least two in each $G_i$, for $i \in \{1, 2, \ldots, \alpha\}$;
(P4) The total degree of every vertex in $G$ is at least three.

**Algorithm.** We give an algorithm for the decision version of the Disjoint SimFVS problem, which only checks whether a solution exists or not. Such an algorithm can be easily modified to find an actual solution if one exists. Let $(G, W, k)$ be an instance of the problem, where $G$ is an $\alpha$-colored graph. If $G[W]$ is not an $\alpha$-forest then we can safely return that $(G, W, k)$ is a no-instance. This follows from the fact that we are looking for an $\alpha$-simfvs in $G$ which is disjoint from $W$. Recall that, in the "compression" step of the iterative compression routine, we always "guess" the intersection of $S^{(j+1)}$ with $Z^{(j+1)}$. Hence, we assume that $G[W]$ is an $\alpha$-forest in what follows. Whenever any of our reduction rules SimFVS.R1 to SimFVS.R5 apply, the algorithm exhaustively does so (in order). If at any point in our algorithm the parameter $k$ drops below zero, then the resulting instance is a no-instance.

**ALGORITHM 1:** Disjoint SimFVS

```
Input: $G = (V, E_1, E_2, \ldots, E_\alpha)$, $W$, $k$, and $\mathfrak{C} = \{C_1, \ldots, C_\alpha\}$
Output: YES if $G$ has an $\alpha$-simfvs $S \subseteq V(G) \setminus W$ of size at most $k$, NO otherwise.

Apply SimFVS R.1 to SimFVS R.5 exhaustively;
if $k < 0$ or for any $i \in \{1, 2, \ldots, \alpha\}$, $|C_i| > k$ then
    return NO
while for some $i \in \{1, 2, \ldots, \alpha\}$, $G_i[V(F_i) \cup V(W_i)]$ is not a forest do
    find a cordate vertex $v_c$ of highest index in some tree of $F_i$;
    Let $u_c, w_c$ be the vertices in tree $T^i_{v_c}$ with a neighbor $u, w$ respectively in $W_i$;
    Let $P = u_c, x_1, \ldots, x_t, v_c$ and $P' = v_c, y_1, \ldots, y_{t'}, w_c$ be the paths in $F_i$ from $u_c$ to $v_c$ and $v_c$ to $w_c$,
      respectively;
    $\mathcal{G}_1 = (G - \{v_c\}, W, k - 1, \mathfrak{C})$, Add $\mathcal{G}_1$ to $\mathfrak{G}$;
    if $V' = V(P) \setminus \{v_c\} \neq \emptyset$ then
        $C_i = C_i \cup \{(u_c, x_1, \ldots, x_t)\}$;
        $\mathcal{G}_2 = (G - V', W, k - 1, \mathfrak{C})$, Add $\mathcal{G}_2$ to $\mathfrak{G}$;
    if $V' = V(P') \setminus \{v_c\} \neq \emptyset$ then
        $C_i = C_i \cup \{(y_1, \ldots, y_{t'}, w_c)\}$;
        $\mathcal{G}_3 = (G - V', W, k - 1, \mathfrak{C})$, Add $\mathcal{G}_3$ to $\mathfrak{G}$;
    if $u, w$ are in the same component of $W_i$ then
        return $\bigvee_{\mathcal{G} \in \mathfrak{G}}$ Disjoint SimFVS($\mathcal{G}$)
    else
        return ($\bigvee_{\mathcal{G} \in \mathfrak{G}}$ Disjoint SimFVS($\mathcal{G}$)) $\vee$ Disjoint SimFVS($G - (V(P) \cup V(P'))$, $W \cup V(P) \cup V(P')$, $k$, $\mathfrak{C}$)
    end
end

// Solve the remaining instance using the hitting set problem
For $i \in \{1, 2, \ldots, \alpha\}$ let $V(C_i) = \bigcup_{C \in C_i} V(C)$, $\mathcal{U} = \bigcup_{i \in \{1,2,\ldots,\alpha\}} V(C_i)$;
$\mathcal{F} = \bigcup_{i \in \{1,2,\ldots,\alpha\}} C_i$;
Find a hitting set $U$ = Hitting Set($\mathcal{F}, \mathcal{U}$);
if $|U| \leq k$ then
    return YES
return NO
```

Recall that initially $F = G - W$ is an $\alpha$-forest, as $W$ is an $\alpha$-simfvs. We will consider each forest $F_i$, for $i \in \{1, 2, \ldots, \alpha\}$, separately (where $F_i$ is the color $i$ graph of the $\alpha$-forest $F$). For $i \in \{1, 2, \ldots, \alpha\}$, we let $W_i = G_i[W] = (W, E_i(G[W]))$ and $\eta_i$ be the number of components in $W_i$. Some of the branching rules that we apply create "special" vertex-disjoint cycles. We will maintain this set of special cycles in $C_i$, for each $i$, and we let $\mathfrak{C} = \{C_1, \ldots, C_\alpha\}$. Initially, $C_i = \emptyset$. Each cycle that we add to $C_i$ will be vertex disjoint from cycles previously added to $C_i$. Hence, if at any point $|C_i| > k$, for any $i$, then we can stop exploring the corresponding branch. Moreover, whenever we "guess" that some vertex $v$ must belong to a solution, we also traverse the family $\mathfrak{C}$ and remove all cycles containing $v$. For the running time analysis of our algorithm we will consider the following measure:

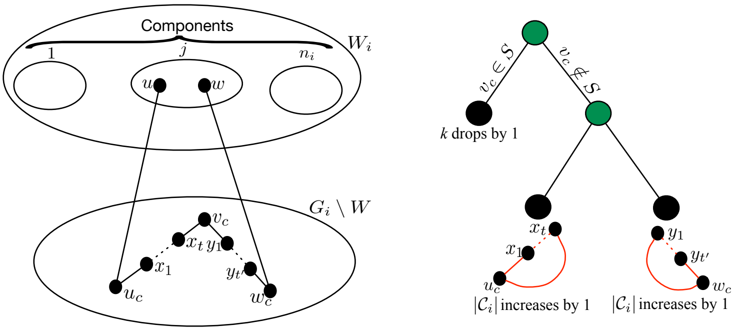

Fig. 1. Branching in Case 1.a

$$\mu = \mu(G, W, k, \mathfrak{C}) = \alpha k + (\sum_{i=1}^{\alpha} \eta_i) - (\sum_{i=1}^{\alpha} |C_i|)$$

The input to our algorithm consists of a tuple $(G, W, k, \mathfrak{C})$. For clarity, we will denote a reduced input by $(G, W, k, \mathfrak{C})$ (the one where reduction rules do not apply).

We root each tree in $F_i$ at some arbitrary vertex. Assign an index $t$ to each vertex $v$ in the forest $F_i$, which is the distance of $v$ from the root of the tree it belongs to (the root is assigned index zero). A vertex $v$ in $F_i$ is called *cordate* if one of the following holds:

- $v$ is a leaf (or degree-zero vertex) in $F_i$ with at least two color $i$ neighbors in $V(W_i)$.
- The subtree $T^i_v$ rooted at $v$ contains two vertices $u$ and $w$ each of which has at least one color $i$ neighbor in $V(W_i)$ ($v$ can be equal to $u$ or $w$).

Lemma 3.4. *For $i \in \{1, 2, \ldots, \alpha\}$, let $v_c$ be a cordate vertex of highest index in some tree of the forest $F_i$ and let $T^i_{v_c}$ denote the subtree rooted at $v_c$. Furthermore, let $u_c$ be one of the vertices in $T^i_{v_c}$ such that $u_c$ has a neighbor in $W_i$. Then, in the path $P = u_c, x_1, \ldots, x_t, v_c$ ($t$ could be equal to zero) between $u_c$ and $v_c$ the vertices $x_1, \ldots, x_t$ are degree-two vertices in $G_i$.*

Proof. If $u_c = v_c$ or $t = 0$ then there is nothing to prove. Otherwise, consider $P = u_c, x_1, \ldots, x_t, v_c$ the path from $v_c$ to $u_c$, where $t \geq 1$. In $P$, if there is a vertex $x$ (other than $u_c$ and $v_c$) which has an edge of color $i$ to a vertex in $W_i$, then $x$ is a cordate vertex of higher index, contradicting the choice of $v_c$. Also, if there is a vertex $x$ in $P$ other than $v_c$ and $u_c$ of degree at least three in $F_i$, the subtree rooted at $x$ has at least two leaves, and all the leaves have a color-$i$ neighbor in $W_i$. Therefore, $x$ is a cordate vertex and has a higher index than $v_c$, contradicting the choice of $v_c$. It follows that $x_1, \ldots, x_t$ (if they exist) are degree-two vertices in $G_i$. □

We consider the following cases depending on whether there is a cordate vertex in $F_i$ or not.

- Case 1: There is a cordate vertex in $F_i$. Let $v_c$ be a cordate vertex with the highest index in some tree in $F_i$ and let the two vertices with neighbors in $W_i$ be $u_c$ and $w_c$ ($v_c$ can be equal to $u_c$ or $w_c$). Let $P = u_c, x_1, x_2, \cdots, x_t, v_c$ and $P' = v_c, y_1, y_2, \cdots, y_{t'}, w_c$ be the unique paths in $F_i$ from $u_c$ to $v_c$ and from $v_c$ to $w_c$, respectively. Let $P_v = u_c, x_1, \cdots, x_t, v_c, y_1, \cdots, y_{t'}, w_c$ be the unique path in $F_i$ from $u_c$ to $w_c$. Consider the following sub-cases:
  - Case 1.a: $u_c$ and $w_c$ have neighbors in the same component of $W_i$. In this case one of the vertices from path $P_v$ must be in the solution (Figure 1). We branch as follows:
    - ∗ $v_c$ belongs to the solution. We delete $v_c$ from $G$ and decrease $k$ by 1. In this branch $\mu$ decreases by $\alpha$.

When $v_c$ does not belong to the solution, then at least one vertex from $u_c, x_1, x_2, \cdots, x_t$ or $y_1, y_2, \cdots, y_{t'}, w_c$ must be in the solution. But note that these are vertices of degree at most two in $G_i$ by Lemma 3.4. So with respect to color $i$, it does not matter which vertex is chosen in the solution. The only issue comes from some color $j$ cycle, where $j \neq i$, in which choosing a particular vertex from $u_c, x_1, \cdots, x_t$ or $y_1, y_2, \cdots, y_{t'}, w_c$ would be more beneficial. We consider the following two cases.

  - ∗ One of the vertices from $u_c, x_1, x_2, \cdots, x_t$ is in the solution. In this case we add an edge $(u_c, x_t)$ (or $(u_c, u_c)$ when $u_c$ and $v_c$ are adjacent) to $G_i$ and delete the edge $(x_t, v_c)$ from $G_i$. This creates a cycle $C$ in $G_i - W$, which is itself a component in $G_i - W$. We remove the edges in $C$ from $G_i$ and add the cycle $C$ to $\mathcal{C}_i$. We will be handling these sets of cycles independently. In this case $|\mathcal{C}_i|$ increases by 1, so the measure $\mu$ decreases by 1.
  - ∗ One of the vertices from $y_1, y_2, \cdots, y_t, w_c$ is in the solution. In this case we add an edge $(y_1, w_c)$ to $G_i$ and delete the edge $(v_c, y_1)$ from $G_i$. This creates a cycle $C$ in $G_i - W$ as a component. We add $C$ to $\mathcal{C}_i$ and delete edges in $C$ from $G_i - W$. In this branch $|\mathcal{C}_i|$ increases by 1, so the measure $\mu$ decreases by 1. The resulting branching vector is $(\alpha, 1, 1)$.

- – Case 1.b: $u_c$ and $w_c$ do not have neighbors in the same component. We branch as follows (Figure 2):
  - ∗ $v_c$ belongs to the solution. We delete $v_c$ from $G$ and decrease $k$ by 1. In this branch $\mu$ decreases by $\alpha$.
  - ∗ One of the vertices from $u_c, x_1, x_2, \cdots, x_t$ is in the solution. In this case we add an edge $(u_c, x_t)$ to $G_i$ and delete the edge $(x_t, v_c)$ from $G_i$. This creates a cycle $C$ in $G_i - W$ as a component. As in Case 1, we add $C$ to $\mathcal{C}_i$ and delete edges in $C$ from $G_i - W$. $|\mathcal{C}_i|$ increases by 1, so the measure $\mu$ decreases by 1.
  - ∗ One of the vertices from $y_1, y_2, \cdots, y_t, w_c$ is in the solution. In this case we add an edge $(y_1, w_c)$ to $G_i$ and delete the edge $(v_c, y_1)$ from $G_i$. This creates a cycle $C$ in $G_i - W$ as a component. We add $C$ to $\mathcal{C}_i$ and delete edges in $C$ from $G_i - W$. In this branch $|\mathcal{C}_i|$ increases by 1, so the measure $\mu$ decreases by 1.
  - ∗ No vertex from path $P_v$ is in the solution. In this case we add the vertices in $P_v$ to $W$, the resulting instance is $(G - P_v, W \cup P_v, k)$. The number of components in $W_i$ decreases and we get a drop of 1 in $\eta_i$, so $\mu$ decreases by 1. Note that if $G[W \cup P_v]$ is not acyclic we can safely ignore this branch. The resulting branching vector is $(\alpha, 1, 1, 1)$.

- • Case 2: There is no cordate vertex in $F_i$. Let $\mathcal{F}$ be a family of sets containing a set $f_C = V(C)$ for each $C \in \bigcup_{i=1}^{\alpha} \mathcal{C}_i$ and let $\mathcal{U} = \bigcup_{i=1}^{\alpha}(\bigcup_{C \in \mathcal{C}_i} V(C))$. Note that $|\mathcal{F}| \leq \alpha k$. We find a subset $U \subseteq \mathcal{U}$ (if it exists) which hits all the sets in $\mathcal{F}$, such that $|U| \leq k$.

Note that in Case 1, if the cordate vertex $v_c$ is a leaf, then $u_c = w_c = v_c$. Therefore, from Case 1.a we are left with one branching rule. Similarly, we are left with the first and the last branching rules for Case 1.b. If $v_c$ is not a leaf but $v_c$ is equal to $u_c$ or $w_c$, say $v_c = w_c$, then for both Case 1.a and Case 1.b we do not have to consider the third branch. Finally, when none of the reduction or branching rules apply, we solve the problem by invoking an algorithm for the Hitting Set problem as a subroutine.

Lemma 3.5. *The presented algorithm for* Disjoint SimFVS *is correct.*

Proof. Consider an input $(G, W, k, \mathfrak{C})$ to the algorithm for Disjoint SimFVS, where $G$ is an $\alpha$-colored graph, $W$ is an $\alpha$-simfvs of size $k+1$, and $k$ is a positive integer and $\mathfrak{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_1\}$. Let $\mu = \mu(G, W, k, \mathfrak{C})$ be the measure as defined earlier. We prove the correctness of the algorithm by induction on the measure $\mu$. The base case occurs when one of the following holds:

- $k < 0$,

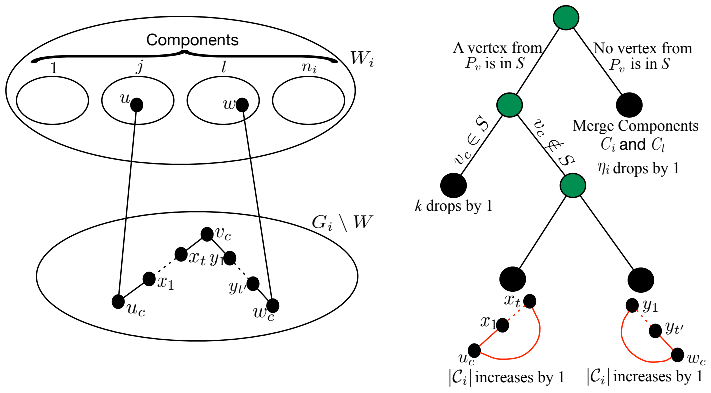

Fig. 2. Branching: Case 1.b

- for some $i \in \{1, 2, \ldots, \alpha\}$, $|C_i| > k$, or
- $\mu \leq 0$.

If $k < 0$, then we can safely conclude that $G$ is a no-instance. If for some $i \in \{1, 2, \ldots, \alpha\}$ we have $|C_i| > k$, then we need to pick at least one vertex from each of the vertex-disjoint cycles in $C_i$ and there are at least $k + 1$ of them. Our algorithm correctly concludes that the graph is also a no-instance in such cases. If $\mu = \alpha k + (\sum_{i=1}^{\alpha} \eta_i) - (\sum_{i=1}^{\alpha} |C_i|) \leq 0$ then $\alpha k \leq \sum_{i=1}^{\alpha} |C_i|$. But for each $i \in \{1, 2, \ldots, \alpha\}$, we have $|C_i| \leq k$. Therefore $\alpha k \leq \sum_{i=1}^{\alpha} |C_i| \leq \alpha k$, $\sum_{i=1}^{\alpha} |C_i| = \alpha k$, and $|C_i| = k$, for all $i \in \{1, 2, \ldots, \alpha\}$. This implies that for each $i \in \{1, 2, \ldots, \alpha\}$, $G_i[V(F_i) \cup V(W_i)]$ must be acyclic. Assume otherwise. Then, for some $i \in \{1, 2, \ldots, \alpha\}$, $G_i[V(F_i) \cup V(W_i)]$ contains a cycle which is vertex disjoint from the $k$ cycles in $C_i$. Therefore, at least $k + 1$ vertices are needed to intersect these cycles and we again have a no-instance. Recall that when a new vertex $v$ is added to the solution set we delete all those cycles in $\bigcup_{i=1}^{\alpha} C_i$ which contain $v$.

We are now left with cycles in $\bigcup_{i=1}^{\alpha} C_i$. Intersecting a cycle $C \in \bigcup_{i=1}^{\alpha} C_i$ is equivalent to hitting the set $V(C)$. Hence, we construct a family $\mathcal{F}$ consisting of a set $f_C = V(C)$ for each $C \in \bigcup_{i=1}^{\alpha} C_i$ and we let $\mathcal{U} = \bigcup_{i=1}^{\alpha} (\bigcup_{C \in C_i} V(C))$. Note that $|\mathcal{F}| \leq \alpha k$. If we can find a subset $U \subseteq \mathcal{U}$ which hits all the sets in $\mathcal{F}$, such that $|U| \leq k$, then $U$ is the required solution. Otherwise, we have a no-instance. It is known that the Hitting Set problem parameterized by the size of the family $\mathcal{F}$ is fixed-parameter tractable and can be solved in $O^\star(2^{|\mathcal{F}|})$ time [7]. In particular, we can find an optimum hitting set $U \subseteq \mathcal{U}$, hitting all the sets in $\mathcal{F}$. Therefore, we have a subset of vertices that intersects all the cycles in $C_i$, for $i \in \{1, 2, \ldots, \alpha\}$.

Putting it all together, at a base case, our algorithm correctly decides whether $(G, W, k, \mathfrak{C})$ is a yes-instance or not. For the induction hypothesis, assume that the algorithm correctly decides an instance for $\mu \leq t$. Now consider the case $\mu = t + 1$. If some reduction rule applies then we create an equivalent instance (since all reduction rules are safe). Therefore, either we get an equivalent instance with the same measure or we get an equivalent instance with $\mu \leq t$ (the case when SimFVS.R5 is applied). In the latter case, by the induction hypothesis, our algorithm correctly decides the instance where $\mu \leq t$. In the former case, we apply one of the branching rules. Each branching rule is exhaustive and covers all possible cases. In addition, the measure decreases at each branch by at least one. Therefore, by the induction hypothesis, the algorithm correctly decides whether the input is a yes-instance or not. □

Lemma 3.6. Disjoint SimFVS *is solvable in time* $O^\star(22^{\alpha k})$.

Proof. All of the reduction rules SimFVS.R1 to SimFVS.R5 can be applied in time polynomial in the input size. Also, at each branch we spend a polynomial amount of time. For each of the recursive calls at a branch, the measure $\mu$ decreases at least by 1. When $\mu \leq 0$, then we are able to solve the remaining instance in time $\mathcal{O}(2^{\alpha k})$ or correctly conclude that the corresponding branch cannot lead to a solution. At the start of the algorithm $\mu \leq 2\alpha k$. Therefore, the height of the search tree is bounded by $2\alpha k$. The worst-case branching vector for the algorithm is $(\alpha, 1, 1, 1)$. The recurrence relation for the worst case branching vector is: $T(\mu) \leq T(\mu - \alpha) + 3T(\mu - 1) \leq T(\mu - 2) + 3T(\mu - 1)$, since $\alpha \geq 2$. The running time corresponding to the above recurrence relation is $3.303^{2\alpha k}$. At each branch we spend a polynomial amount of time but we might require $\mathcal{O}(2^{\alpha k})$ time for solving the base case. Therefore, the running time of the algorithm is $\mathcal{O}^\star(2^{\alpha k} \cdot 3.303^{2\alpha k}) = \mathcal{O}^\star(22^{\alpha k})$. □

Theorem 3.7. Simultaneous Feedback Vertex Set *is solvable in time* $\mathcal{O}^\star(23^{\alpha k})$.

### 3.2 Faster algorithm for Simultaneous Feedback Vertex Set with $\alpha = 2$

We improve the running time of the FPT algorithm for SimFVS when $\alpha = 2$. Given two sets of disjoint cycles $\mathcal{C}_1$ and $\mathcal{C}_2$ and a set $V = \bigcup_{C \in \mathcal{C}_1 \cup \mathcal{C}_2} V(C)$, we want to find a subset $H \subseteq V$ such that $H$ contains at least one vertex from $V(C)$, for each $C \in \mathcal{C}_1 \cup \mathcal{C}_2$. We construct a bipartite graph $G_M$ as follows. We set $V(G_M) = \{c_x^1 | C_x \in \mathcal{C}_1\} \cup \{c_y^2 | C_y \in \mathcal{C}_2\}$. In other words, we create one vertex for each cycle in $\mathcal{C}_1 \cup \mathcal{C}_2$. We add an edge between $c_x^1$ and $c_y^2$ if and only if $V(C_x) \cap V(C_y) \neq \emptyset$. Note that for $i \in \{1, 2\}$ and $C, C' \in \mathcal{C}_i$, $V(C) \cap V(C') = \emptyset$. In Lemma 3.8, we show that finding a matching $M$ in $G_M$, such that $|M| + |V(G_M) \setminus V(M)| \leq k$ (or equivalently $|\mathcal{C}_1| + |\mathcal{C}_2| - k \leq |M|$), corresponds to finding a set $H$ of size at most $k$, such that $H$ contains at least one vertex from each cycle $C \in \mathcal{C}_1 \cup \mathcal{C}_2$.

Lemma 3.8. *For $i \in \{1, 2\}$, let $\mathcal{C}_i$ be a set of vertex-disjoint cycles, i.e. for each $C, C' \in \mathcal{C}_i$, $C \neq C'$ implies $V(C) \cap V(C') = \emptyset$. Let $\mathcal{F} = \{V(C) | C \in \mathcal{C}_1 \cup \mathcal{C}_2\}$ and $\mathcal{U} = \bigcup_{C \in \mathcal{C}_1 \cup \mathcal{C}_2} V(C)$. There exists a vertex subset $H \subseteq \mathcal{U}$ of size $k$ such that $H \cap V(C) \neq \emptyset$, for each $C \in \mathcal{C}_1 \cup \mathcal{C}_2$, if and only if $G_M$ has a matching $M$, such that $|M| + |V(G_M) \setminus V(M)| \leq k$.*

Proof. For the forward direction, consider a minimal vertex subset $H \subseteq V(\mathcal{C}_1) \cup V(\mathcal{C}_2)$ of size at most $k$ such that for each $C \in \mathcal{C}_1 \cup \mathcal{C}_2$, $H \cap V(C) \neq \emptyset$. Note that a vertex $h \in H$ can be present in at most one cycle from $\mathcal{C}_i$, for $i \in \{1, 2\}$, since $\mathcal{C}_i$ is a set of vertex-disjoint cycles. Therefore, $h$ can be present in at most 2 cycles from $\mathcal{C}_1 \cup \mathcal{C}_2$. If $h$ is present in 2 cycles, say $C_x \in \mathcal{C}_1$ and $C_y \in \mathcal{C}_2$, then in $G_M$ we must have an edge between $c_x^1$ and $c_y^2$ (since $h$ belongs to both $C_x$ and $C_y$). We include the edge $(c_x^1, c_y^2)$ in the matching $M$. If $h$ belongs to only one cycle, say $C_z^i \in \mathcal{C}_1 \cup \mathcal{C}_2$, then we include vertex $c_z^i$ in a set $I$. Note that $(V(G_M) \setminus V(M)) \subseteq I$. For each $h \in H$, we either add a matching edge or add a vertex to $I$. Therefore $|M| + |V(G_M) \setminus V(M)| \leq |M| + |I| \leq k$.

In the reverse direction, consider a matching $M$ such that $|M| + |V(G_M) \setminus V(M)| \leq k$. We construct a set $H$ of size at most $k$ containing a vertex from each cycle in $\mathcal{C}_1 \cup \mathcal{C}_2$. For each edge $(c_x^1, c_y^2)$ in the matching, where $C_x \in \mathcal{C}_1$ and $C_y \in \mathcal{C}_2$, there is a vertex $h$ that belongs to both $V(C_x)$ and $V(C_y)$. Include $h$ in $H$. For each $c_z^i \in V(G_M) \setminus V(M)$, add an arbitrary vertex $v \in V(C_z)$ to $H$. Note that $|H| \leq k$, since for each matching edge and each unmatched vertex we added one vertex to $H$. Moreover, for each cycle $C \in \mathcal{C}_1 \cup \mathcal{C}_2$, its corresponding vertex in $G_M$ is either part of the matching or is an unmatched vertex; in both cases there is a vertex in $H$ that belongs to $C$. Therefore, $H$ is a subset of size at most $k$ which contains at least one vertex from each cycle in $\mathcal{C}_1 \cup \mathcal{C}_2$. □

Note that a matching $M$ in $G_M$ minimizing $|M| + |V(G_M) \setminus V(M)|$ is one of maximum size. Therefore, at the base case, we compute a maximum matching of the corresponding graph $G_M$, which is a polynomial-time solvable problem, and return an optimal solution for intersecting all cycles in $C_1 \cup C_2$. Moreover, if we set $\mu = 2k + (\eta_1/2 + \eta_2/2) - (|C_1| + |C_2|)$, then the worst case branching vector is $(2, 1, 1, 1/2)$. Corresponding to this worst case branching vector, the running time of the algorithm is $O^\star(81^k)$.

Theorem 3.9. Simultaneous Feedback Vertex Set *is solvable in time* $O^\star(81^k)$ *when* $\alpha = 2$.

## 4 POLYNOMIAL KERNEL FOR SIMULTANEOUS FEEDBACK VERTEX SET

In this section we give a kernel with $O(\alpha k^{3(\alpha+1)})$ vertices for SimFVS. Let $(G, k)$ be an instance of SimFVS, where $G$ is an $\alpha$-colored graph and $k$ is a positive integer. We assume that reduction rules SimFVS.R1 to SimFVS.R5 have been exhaustively applied. The kernelization algorithm then proceeds in two stages. In stage one, we bound the maximum degree of each $G_i$ (hence bounding the total degree of $G$). In the second stage, we present new reduction rules to deal with degree-two vertices and conclude a bound on the total number of vertices.

To bound the total degree of each vertex $v \in V(G)$, we bound the degree of $v$ in $G_i$, for $i \in \{1, 2, \ldots, \alpha\}$. To do so, we need the Expansion Lemma [7] as well as the 2-approximation algorithm for the classical Feedback Vertex Set problem [1].

A *q-star*, $q \geq 1$, is a graph with $q + 1$ vertices, one vertex of degree $q$ and all other vertices of degree 1. Let $H$ be a bipartite graph with vertex bipartition $(A, B)$. A set of edges $M \subseteq E(H)$ is called a *q-expansion* of $A$ into $B$ if (i) every vertex of $A$ is incident with exactly $q$ edges of $M$ and (ii) exactly $q|A|$ vertices in $B$ are incident with edges of $M$.

Lemma 4.1 (Expansion Lemma [7]). *Let $q$ be a positive integer and $H$ be a bipartite graph with vertex bipartition $(A, B)$ such that $|B| \geq q|A|$ and there are no isolated vertices in $B$. Then, there exist nonempty vertex sets $X \subseteq A$ and $Y \subseteq B$ such that:*

(1) *$X$ has a $q$-expansion into $Y$ and*
(2) *no vertex in $Y$ has a neighbour outside $X$, i.e. $N(Y) \subseteq X$.*

*Furthermore, the sets $X$ and $Y$ can be found in time polynomial in the size of $H$.*

### 4.1 Bounding the degree of vertices in $G_i$

We now describe the reduction rules that allow us to bound the maximum degree of a vertex $v \in V(G)$.

Lemma 4.2 (Lemma 6.8 [25]). *Let $G$ be an undirected (multi) graph and $x$ be a vertex of $G$ without a loop. Then in polynomial time we can either decide that $(G, k)$ is a no-instance of* Feedback Vertex Set *or check whether there is an $x$-flower of order $k + 1$, or find a set of vertices $Z \subseteq V(G) \setminus \{x\}$ of size at most $3k$ intersecting every cycle in $G$, i.e. $Z$ is a feedback vertex set of $G$.*

The next proposition easily follows from Lemma 4.2.

Proposition 4.3. *Let $G$ be an undirected $\alpha$-colored multigraph and $x$ be a vertex without a loop in $G_i$, for some $i \in \{1, 2, \ldots, \alpha\}$. Then in polynomial time we can either decide that $(G, k)$ is a no-instance of* Simultaneous Feedback Vertex Set *or check whether there is an $x$-flower of order $k + 1$ in $G_i$, or find a set of vertices $Z \subseteq V(G) \setminus \{x\}$ of size at most $3k$ intersecting every cycle in $G_i$.*

After applying reduction rules SimFVS.R1 to SimFVS.R5 exhaustively, we know that the degree of a vertex in each $G_i$ is either 0 or at least 2 and no vertex has a loop. Now consider a vertex $v$ whose degree in $G_i$ is more than $3k(k + 4)$. By Proposition 4.3, we know that one of three cases must apply:

(1) $(G, k)$ is a no-instance of SimFVS,
(2) we can find (in polynomial time) a $v$-flower of order $k + 1$ in $G_i$, or

(3) we can find (in polynomial time) a set $H_v \subseteq V(G_i)$ of size at most $3k$ such that $v \notin H_v$ and $G_i - H_v$ is a forest.

The following reduction rule allows us to deal with case (2). The safeness of the rule follows from the fact that if $v$ in not included in the solution then we need to have at least $k + 1$ vertices in the solution.

Reduction SimFVS.R6. For $i \in \{1, 2, \ldots, \alpha\}$, if $G_i$ has a vertex $v$ such that there is a $v$-flower of order at least $k+1$ in $G_i$, then include $v$ in the solution $S$ and decrease $k$ by 1. The resulting instance is $(G - \{v\}, k - 1)$.

When in case (3), we bound the degree of $v$ as follows. Consider the graph $G'_i = G_i - (H_v \cup \{v\} \cup V^i_0)$, where $V^i_0$ is the set of degree-zero vertices in $G_i$. Let $\mathcal{D}$ be the set of components in the graph $G'_i$ which have a vertex adjacent to $v$. Note that each $D \in \mathcal{D}$ is a tree and $v$ cannot have two neighbors in $D$, since $H_v$ is a feedback vertex set in $G_i$. We will now argue that each component $D \in \mathcal{D}$ has a vertex $u$ such that $u$ is adjacent to a vertex in $H_v$. Suppose for a contradiction that there is a component $D \in \mathcal{D}$ such that $D$ has no vertex which is adjacent to a vertex in $H_v$. $D \cup \{v\}$ is a tree with at least 2 vertices, so $D$ has a vertex $w$, such that $w$ is a degree-one vertex in $G_i$, contradicting the fact that each vertex in $G_i$ is either of degree zero or of degree at least two.

After exhaustive application of SimFVS.R4, every pair of vertices in $G_i$ can have at most two edges between them. In particular, there can be at most two edges between $h \in H_v$ and $v$. If the degree of $v$ in $G_i$ is more than $3k(k + 4)$, then the number of components $|\mathcal{D}|$ in $G'_i$ is more than $3k(k + 2)$, since $|H_v| \leq 3k$.

Consider the bipartite graph $\mathcal{B}$, with bipartition $(H_v, Q)$, where $Q$ has a vertex $q_D$ corresponding to each component $D \in \mathcal{D}$. We add an edge between $h \in H_v$ and $q_D \in Q$ to $E(\mathcal{B})$ if and only if $D$ has a vertex $d$ which is adjacent to $h$ in $G_i$.

Reduction SimFVS.R7. Let $v$ be a vertex of degree at least $3k(k+4)$ in $G_i$, for some $i \in \{1, 2, \ldots, \alpha\}$, and let $H_v$ be a feedback vertex set in $G_i$ not containing $v$ and of size at most $3k$.

- Let $Q' \subseteq Q$ and $H \subseteq H_v$ be the sets of vertices obtained after applying Lemma 4.1 with $q = k + 2$, $A = H_v$, and $B = Q$, such that $H$ has a $(k + 2)$-expansion into $Q'$ in $\mathcal{B}$;
- Delete all the edges $(d, v)$ in $G_i$, where $d \in V(D)$ and $q_D \in Q'$;
- Add double edges between $v$ and $h$ in $G_i$, for all $h \in H$ (unless such edges already exist).

By Lemma 4.1 and Proposition 4.3, SimFVS.R7 can be applied in time polynomial in the input size.

Lemma 4.4. *Reduction rule* SimFVS.*R7 is safe.*

Proof. Let $G$ be an $\alpha$-colored graph where reductions SimFVS.R1 to SimFVS.R6 do not apply. Let $v$ be a vertex of degree more than $3k(k + 4)$ in $G_i$, for $i \in \{1, 2, \ldots, \alpha\}$. Let $H \subseteq H_v$, $Q' \subseteq Q$ be the sets defined above and let $G'$ be the instance obtained after a single application of reduction rule SimFVS.R7. We show that $G$ has an $\alpha$-simfvs of size at most $k$ if and only if $G'$ has an $\alpha$-simfvs of size at most $k$. We need the following claim.

Claim 1. *Any $k$-sized $\alpha$-simfvs $S$ of $G$ or $G'$ either contains $v$ or contains all the vertices in $H$.*

Proof. Since there exists a cycle (double edge) between $v$ and every vertex $h \in H$ in $G'_i$, it easily follows that either $v$ or all vertices in $H$ must be in any solution for $G'$.

Consider the case of $G$. We assume $v \notin S$ and there is a vertex $h \in H$ such that $h \notin S$. Note that $H$ has a $(k + 2)$-expansion into $Q'$ in $\mathcal{B}$, therefore $h$ is the center of a $(k + 2)$-star in $\mathcal{B}[H \cup Q']$. Let $Q_h$ be the set of neighbors of $h$ in $\mathcal{B}[H \cup Q']$ ($|Q_h| \geq k + 2$). For each $q_D, q_{D'} \in Q_h$, their corresponding components $D, D' \in \mathcal{D}$ form a cycle with $v$ and $h$. If both $h$ and $v$ are not in $S$, then we need to pick at least $k + 1$ vertices to intersect the cycles formed by $D$, $D'$, $h$, and $v$, for each $q_D, q_{D'} \in Q'$. Therefore, $H \subseteq S$, as needed. □

In the forward direction, consider an $\alpha$-simfvs $S$ of size at most $k$ in $G$. For $j \in \{1, 2, \ldots, \alpha\} \setminus \{i\}$, $G'_j = G_j$ and therefore $S$ intersects all the cycles in $G'_j$. By the previous claim, we can assume that either $v \in S$ or $H \subseteq S$. In both cases, $S$ intersects all the new cycles created in $G'_i$ by adding double edges between $v$ and $h \in H$. Moreover, apart from the double edges between $v$ and $h \in H$, all the cycles in $G'_i$ are also cycles in $G_i$, therefore $S$ intersects all those cycles in $G'_i$. It follows that $S$ is an $\alpha$-simfvs in $G'$.

In the reverse direction, consider an $\alpha$-simfvs $S$ in $G'$ of size at most $k$. Note that for $j \in \{1, 2, \ldots, \alpha\} \setminus \{i\}$, $G'_j = G_j$. Therefore $S$ intersects all the cycles in $G_j$. By the previous claim, at least one of the following must hold: (1) $v \in S$ or (2) $H \subseteq S$. Suppose that (1) $v \in S$ holds. Since $G'_i - \{v\} = G_i - \{v\}$, $S \setminus \{v\}$ intersects all the cycles in $G'_i - \{v\}$ and $G_i - \{v\}$. Therefore $S$ intersects all the cycles in $G_i$ and $S$ is an $\alpha$-simfvs in $G$. In case (2), i.e. when $v \notin S$ but $H \subseteq S$, any cycle in $G$ which does not intersect with $H$ is also a cycle in $G'$ (since such a cycle does not intersect with $H$ and the only deleted edges from $G'$ belong to cycles passing through $H$). In other words, $S \setminus H$ intersects all cycles in both $G'_i - H$ and $G_i - H$ and, consequently, $S$ is an $\alpha$-simfvs in $G$. □

After exhaustively applying all reductions SimFVS.R1 to SimFVS.R7, the degree of a vertex $v \in V(G_i)$ is at most $3k(k + 4) - 1$ in $G_i$, for each $i \in \{1, 2, \ldots, \alpha\}$.

### 4.2 Bounding the number of vertices in $G$

Having bounded the maximum total degree of a vertex in $G$, we now focus on bounding the number of vertices in the entire graph. To do so, we first compute an approximate solution for the SimFVS instance using the polynomial-time 2-approximation algorithm of Bafna et al. [1] for the Feedback Vertex Set problem in undirected graphs. In particular, we compute a 2-approximate solution $S_i$ in $G_i$, for $i \in \{1, 2, \ldots, \alpha\}$. We let $S = \bigcup_{i=1}^{\alpha} S_i$. Note that $S$ is an $\alpha$-simfvs in $G$ and has size at most $2\alpha|S_{OPT}|$, where $|S_{OPT}|$ is an optimal $\alpha$-simfvs in $G$. Let $F_i = G_i - S_i$. Let $T^i_{\leq 1}$, $T^i_2$, and $T^i_{\geq 3}$, be the sets of vertices in $F_i$ having degree at most one in $F_i$, degree exactly two in $F_i$, and degree greater than two in $F_i$, respectively.

Later, we shall prove that bounding the maximum degree in $G$ is sufficient for bounding the sizes of $T^i_{\leq 1}$ and $T^i_{\geq 3}$, for all $i \in \{1, 2, \ldots, \alpha\}$. We now focus on bounding the size of $T^i_2$ which, for each $i \in \{1, 2, \ldots, \alpha\}$, corresponds to a set of degree-two paths. In other words, for a fixed $i$, the graph induced in $F_i$ by the vertices in $T^i_2$, i.e. $F_i[T^i_2]$, is a set of vertex-disjoint paths. We say a set of distinct vertices $\{v_1, \ldots, v_\ell\}$ in $T^i_2$ forms a *maximal degree-two path* if $(v_j, v_{j+1})$ is an edge, for all $1 \leq j < \ell$, and all vertices $\{v_1, \ldots, v_\ell\}$ have degree exactly two in $G_i$.

We enumerate all the maximal degree-two paths in $G_i - S_i$, for $i \in \{1, 2, \ldots, \alpha\}$. Let this set of paths in $G_i - S_i$ be $\mathcal{P}_i = \{P^i_1, P^i_2, \ldots, P^i_{n_i}\}$, where $n_i$ is the number of maximal degree-two paths in $G_i - S_i$. We introduce a special symbol $\phi$ and add $\phi$ to each set $\mathcal{P}_i$, for $i \in \{1, 2, \ldots, \alpha\}$. The special symbol will be used later to indicate that no path is chosen from the set $\mathcal{P}_i$.

Let $\mathfrak{S} = \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_\alpha$ be the set of all tuples of maximal degree-two paths of different colors. For $\tau \in \mathfrak{S}$, $j \in \{1, 2, \ldots, \alpha\}$, $j(\tau)$ denotes the element from the set $\mathcal{P}_j$ in the tuple $\tau$, i.e. for $\tau = (Q_1, \phi, \ldots, Q_j, \ldots, Q_\alpha)$, $j(\tau) = Q_j$ (for example $2(\tau) = \phi$).

For a maximal degree-two path $P^i_j \in \mathcal{P}_i$ and $\tau \in \mathfrak{S}$, we define $\mathit{Intercept}(P^i_j, \tau) = \emptyset$ if $P^i_j \notin \tau$. Otherwise, we define $\mathit{Intercept}(P^i_j, \tau) = \{v \in V(P^i_j) |$ for all $1 \leq t \leq \alpha$, if $t(\tau) \neq \phi$ then $v \in V(t(\tau))\}$. Stated differently, $\mathit{Intercept}(P^i_j, \tau)$ is either the empty set or is the set of vertices which are present in all the paths in the tuple $\tau$ (of course a $\phi$ entry does not contribute to this set).

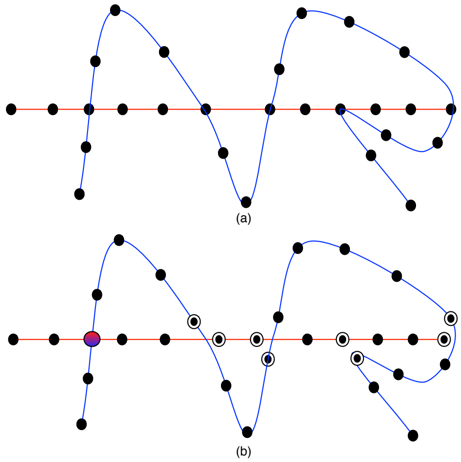

Fig. 3. Unravelling two paths with five common vertices (a) to obtain two paths with one common vertex (b).

We define the notion of *unravelling* a path $P^i_j \in \mathcal{P}_i$ from all other paths of different colors in $\tau \in \mathfrak{S}$ at a vertex $u \in \mathit{Intercept}(P^i_j, \tau)$ by creating a separate copy of $u$ for each path. Formally, for a path $P^i_j \in \mathcal{P}_i$, $\tau \in \mathfrak{S}$, and a vertex $u \in \mathit{Intercept}(P^i_j, \tau)$, the $\mathit{Unravel}(P^i_j, \tau, u)$ operation does the following. For each $t \in \{1, 2, \ldots, \alpha\}$ let $x_t$ and $y_t$ be the unique neighbors of $u$ on path $t(\tau)$ (and also in $G_i$). Create a vertex $u_{t(\tau)}$ for each path $t(\tau)$, for $1 \leq t \leq \alpha$, delete the edges $(x_t, u)$ and $(u, y_t)$ from $G_t$ and add the edges $(x_t, u_{t(\tau)})$ and $(u_{t(\tau)}, y_t)$ in $G_t$. Figure 3 illustrates the unravel operation for two paths of different colors.

Reduction SimFVS.R8. For a path $P^i_j \in \mathcal{P}_i$, $\tau \in \mathfrak{S}$, if $|\mathit{Intercept}(P^i_j, \tau)| > 1$, then for a vertex $u \in \mathit{Intercept}(P^i_j, \tau)$, $\mathit{Unravel}(P^i_j, \tau, u)$.

Lemma 4.5. *Reduction rule* SimFVS.*R8 is safe.*

Proof. Let $G$ be an $\alpha$-colored graph and $S_i$ be a 2-approximate feedback vertex set in $G_i$, for $i \in \{1, 2, \ldots, \alpha\}$. Let $\mathcal{P}_i$ be the set of maximal degree-two paths in $G_i - S_i$ and $\mathfrak{S} = \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_\alpha$. For a path $P^i_j \in \mathcal{P}_i$, $\tau \in \mathfrak{S}$, $|\mathrm{Intercept}(P^i_j, \tau)| > 1$, and $u \in \mathrm{Intercept}(P^i_j, \tau)$, let $G'$ be the $\alpha$-colored graph obtained after applying $\mathrm{Unravel}(P^i_j, \tau, u)$ in $G$. We show that $G$ has an $\alpha$-simfvs of size at most $k$, if and only if $G'$ has an $\alpha$-simfvs of size at most $k$.

In the forward direction, consider an $\alpha$-simfvs $S$ in $G$ of size at most $k$. Let $x$ be a vertex in $\mathit{Intercept}(P^i_j, \tau) \setminus \{u\}$. We define $S' = S$ if $u \notin S$ and $S' = (S \setminus \{u\}) \cup \{x\}$ otherwise. A cycle $C$ in the graph $G'_t$ not containing $u_{t(\tau)}$, where $u_{t(\tau)}$ is the copy of $u$ created for path $t(\tau)$, $\tau \in \mathfrak{S}$, and $t \in \{1, 2, \ldots, \alpha\}$, is also a cycle in $G_t$. Therefore $S'$ intersects $C$. Let $P_t$ be the path in $\mathcal{P}_t$ containing $u$, for $t \in \{1, 2, \ldots, \alpha\}$. Note that in $\mathcal{P}_i$, there is exactly one maximal degree-two path containing $u$ and all the cycles in $G_t$ containing $u$ must contain $P_t$. All the cycles in $G'_t$ containing $u_{t(\tau)}$ must contain $x$, since $u_{t(\tau)}$ is the private copy of $u$ for the degree-two path $t(\tau)$ containing $x$. We consider the following cases depending on whether $u$ belongs to $S$ or not.

- $u \in S$: A cycle $C$ in $G'_t$, $t \in \{1, 2, \ldots, \alpha\}$, containing $u_{t(\tau)}$ also contains $x$. Therefore $S'$ intersects $C$.

- $u \notin S$: Corresponding to a cycle $C$ in $G'_t$, $t \in \{1, 2, \ldots, \alpha\}$, containing $u_{t(\tau)}$, there is a cycle $C'$ on vertices $(V(C) \cup \{u\}) \setminus \{u_{t(\tau)}\}$ in $G_t$. But $S$ is an $\alpha$-simfvs in $G$ and therefore both $S$ and $S'$ must contain a vertex $y \in V(C') \setminus \{u\}$.

In the reverse direction, let $S$ be an $\alpha$-simfvs in $G'$. We define $S' = S$ if $\{u_{l(\tau)} | u_{l(\tau)} \in S, 1 \leq l \leq \alpha\} \cap S \neq \emptyset$ and $S' = (S \setminus \{u_{l(\tau)} | u_{l(\tau)} \in S, 1 \leq l \leq \alpha\}) \cup \{u\}$ otherwise. All the cycles in $G_t$ not containing $u$ are the cycles in $G'_t$ not containing $u_{t(\tau)}$. Therefore $S'$ intersects all those cycles. We consider the following cases depending on whether there is some $t' \in \{1, 2, \ldots, \alpha\}$ for which $u_{t'(\tau)}$ belongs to $S$ or not.

- For all $t' \in \{1, 2, \ldots, \alpha\}$, $u_{t'(\tau)} \notin S$. Let $C$ be a cycle in $G_t$ containing $u$, for $t \in \{1, 2, \ldots, \alpha\}$. Note that $G'_t$ has a cycle $C'$ corresponding to $C$, with $V(C') = (V(C) \setminus \{u\}) \cup \{u_{t(\tau)}\}$. $S$ intersects $C'$, therefore both $S$ and $S'$ have a vertex $y \in V(C') \setminus \{u_{t(\tau)}\}$. Since $y \in V(C)$, $S'$ intersects the cycle $C$ in $G_t$.
- For some $t' \in \{1, 2, \ldots, \alpha\}$, $u_{t'(\tau)} \in S$. Note that $S'$ intersects all the cycles in $G_t$ containing $u$, for $t \in \{1, 2, \ldots, \alpha\}$. Moreover, the only purpose of $u_{t'(\tau)}$ being in $S$ is to intersect a cycle $C'$ in $G'_t$ containing $u_{t'(\tau)}$. However, the corresponding cycle in $G_t$ can be intersected by a single vertex, namely $u$. Therefore, $S'$ is an $\alpha$-simfvs in $G$.

This completes the proof. □

Theorem 4.6. SimFVS *admits a kernel on* $O(\alpha k^{3(\alpha+1)})$ *vertices.*

Proof. Consider an $\alpha$-colored graph $G$ on which reduction rules SimFVS.R1 to SimFVS.R8 have been exhaustively applied. Note that all of our reduction rules are safe. Reduction rules SimFVS.R1 to SimFVS.R5 can clearly be applied in time polynomial in $|V(G)|$ and $k$ (for constant $\alpha$). The fact that we can apply reduction rules SimFVS.R6 and SimFVS.R7 in polynomial time (for constant $\alpha$) follows from Lemma 4.1 and Proposition 4.3. Moreover, observe that reduction rules SimFVS.R1 to SimFVS.R7 strictly decrease either the number of vertices or the number of edges in the graph, and therefore can only be applied a polynomial number of times. As for SimFVS.R8, we shall show in what follows that the number of maximal degree-two paths is bounded by $k^{O(\alpha)}$ (assuming SimFVS.R1 to SimFVS.R7 have been exhaustively applied), and we can enumerate all of them in polynomial time (for constant $\alpha$). Even though SimFVS.R8 increases the number of vertices in the graph, such vertices will always have degree exactly two in the graph and will subsequently be removed. Every application of SimFVS.R8 decreases the number of vertices sharing the same set of maximal degree-two paths (and no reduction rule increases the number of such vertices). Hence, It remains to bound the number of vertices.

For $i \in \{1, 2, \ldots, \alpha\}$, the degree of a vertex $v \in G_i$ is either 0 or at least 2 in $G_i$. In what follows, we do not count the vertices of degree 0 in $G_i$ while counting the vertices in $G_i$; since the total degree of a vertex $v \in V(G)$ is at least three, there is some $j \in \{1, 2, \ldots, \alpha\}$ such that the degree of $v \in V(G_j)$ is at least 2.

Let $S_i$ be a 2-approximate feedback vertex set in $G_i$, for $i \in \{1, 2, \ldots, \alpha\}$. Note that $S = \bigcup_{i=1}^{\alpha} S_i$ is a $2\alpha$-approximate $\alpha$-simfvs in $G$. Let $F_i = G_i - S_i$. Let $T^i_{\leq 1}$, $T^i_2$, and $T^i_{\geq 3}$, be the sets of vertices in $F_i$ having degree at most one in $F_i$, degree exactly two in $F_i$, and degree greater than two in $F_i$, respectively. The degree of each vertex $v \in V(G_i)$ is bounded by $O(k^2)$ in $G_i$, for $i \in \{1, 2, \ldots, \alpha\}$. In particular, the degree of each $s \in S$ is bounded by $O(k^2)$ in $G_i$. Moreover, each vertex $v \in T^i_{\leq 1}$ has degree at least 2 in $G_i$ and must therefore be adjacent to some vertex in $S$. It follows that $|T^i_{\leq 1}| \in O(k^3)$.

In a tree, the number $t$ of vertices of degree at least three is bounded by $l - 2$, where $l$ is the number of leaves. Hence, $|T^i_{\geq 3}| \in O(k^3)$. Also, in a tree, the number of maximal degree-two paths is

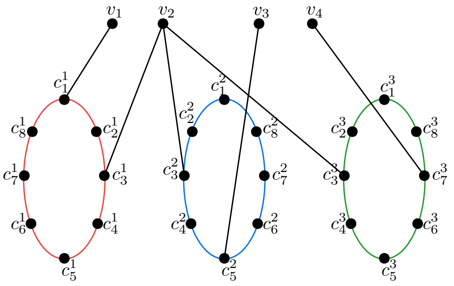

Fig. 4. The graph $G$ before contracting all edges colored zero for $\mathcal{U} = \{u_1, u_2, u_3, u_4\}$ and $\mathcal{F} = \{\{u_1, u_2\}, \{u_2, u_3\}, \{u_2, u_4\}\}$.

bounded by $t + l$. Consequently, the number of degree-two paths in $G_i - S_i$ is in $O(k^3)$. Moreover, no two maximal degree-two paths in a tree intersect.

Note that there are at most $O(k^3)$ maximal degree-two paths in $\mathcal{P}_i$, for $i \in \{1, 2, \ldots, \alpha\}$, and therefore $|\mathfrak{S}| = O(k^{3\alpha})$. After exhaustive application of SimFVS.R8, for each path $P^i_j \in \mathcal{P}_i$, $i \in \{1, 2, \ldots, \alpha\}$, and $\tau \in \mathfrak{S}$, there is at most one vertex in $Intercept(P^i_j, \tau)$. Also note that after exhaustive application of reductions SimFVS.R1 to SimFVS.R7, the total degree of a vertex in $G$ is at least 3. Therefore, there can be at most $O(k^{3\alpha})$ vertices in a degree-two path $P^i_j \in \mathcal{P}_i$. Furthermore, there are at most $O(k^3)$ degree-two maximal paths in $G_i$, for $i \in \{1, 2, \ldots, \alpha\}$. It follows that $|T^i_2| \in O(k^{3(\alpha+1)})$ and $|V(G_i)| \leq |T^i_{\leq 1}| + |T^i_2| + |T^i_{\geq 3}| + |S_i| = O(k^3) + O(k^{3(\alpha+1)}) + O(k^3) + 2k \in O(k^{3(\alpha+1)})$. Therefore, the number of vertices in $G$ is in $O(\alpha k^{3(\alpha+1)})$. □

## 5 HARDNESS RESULTS

In this section we show that SimFVS is W[1]-hard when $\alpha \in O(\log n)$, where $n$ is the number of vertices in the input graph. We give a reduction from a special version of the Hitting Set (HS) problem, which we denote by Partitioned Hitting Set (PHS). We believe this version of Hitting Set to be of independent interest with possible applications for showing hardness results of similar flavor. We prove W[1]-hardness of Partitioned Hitting Set by a reduction from a restricted version of the Partitioned Subgraph Isomorphism (PSI) problem.

Before we delve into the details, we start with a simpler reduction from Hitting Set showing that SimFVS is W[2]-hard when $\alpha \in O(n)$. The reduction closely follows that of Lokshtanov [20] for dealing with the Wheel-Free Deletion problem. Intuitively, starting with an instance $(\mathcal{U}, \mathcal{F}, k)$ of HS, we first construct a graph $G$ on $2|\mathcal{U}||\mathcal{F}|$ vertices consisting of $|\mathcal{F}|$ vertex-disjoint cycles. Then, we use $|\mathcal{F}|$ colors to uniquely map each set to a separate cycle; carefully connecting these cycles together guarantees equivalence of both instances.

Theorem 5.1. SimFVS *parameterized by solution size is W[2]-hard when $\alpha \in O(n)$.*

Proof. Given an instance $(\mathcal{U}, \mathcal{F}, k)$ of Hitting Set, we let $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$ and $\mathcal{F} = \{f_1, \ldots, f_{|\mathcal{F}|}\}$. We assume, without loss of generality, that each element in $\mathcal{U}$ belongs to at least one set in $\mathcal{F}$. For each $f_i \in \mathcal{F}$, $1 \leq i \leq |\mathcal{F}|$, we create a vertex-disjoint cycle $C_i$ on $2|\mathcal{U}|$ vertices and assign all its edges color $i$. We let $V(C_i) = \{c^i_1, c^i_2, \ldots, c^i_{2|\mathcal{U}|}\}$ and we define $\beta(i, u_j) = c^i_{2j-1}$, for $1 \leq i \leq |\mathcal{F}|$ and $1 \leq j \leq |\mathcal{U}|$. In other words, every odd-numbered vertex of $C_i$ is mapped to an element in $\mathcal{U}$. Now for every element $u_j \in \mathcal{U}$, $1 \leq j \leq |\mathcal{U}|$, we create a vertex $v_j$, we let $\gamma(u_j) = \{c^i_{2j-1} | 1 \leq i \leq |\mathcal{F}| \wedge u_j \in f_i\}$, and we add an edge (of some special color, say zero) between $v_j$ and every vertex in $\gamma(u_j)$ (see Figure 4).

To finalize the reduction, we contract all the edges colored zero to obtain an instance $(G, k)$ of SimFVS. Note that $|V(G)| = |E(G)| = 2|\mathcal{U}||\mathcal{F}|$ and the total number of used colors is $|\mathcal{F}|$ (and $|\mathcal{F}| \in O(|\mathcal{U}||\mathcal{F}|)$). Moreover, after contracting all 0-colored edges, $|\gamma(u_j)| = 1$ for all $u_j \in \mathcal{U}$.

Claim 2. *If $\mathcal{F}$ admits a hitting set of size at most $k$ then $G$ admits an $|\mathcal{F}|$-simfvs of size at most $k$.*

Proof. Let $X = \{u_{i_1}, \ldots, u_{i_k}\}$ be such a hitting set. We construct a vertex set $Y = \{\gamma(u_{i_1}), \ldots, \gamma(u_{i_k})\}$. Recall that, after contracting all 0-colored edges, $|\gamma(u_j)| = 1$ for all $u_j \in \mathcal{U}$, and therefore the vertex set $Y$ is well-defined. If $Y$ is not an $|\mathcal{F}|$-simfvs of $G$ then $G[V(G) \setminus Y]$ must contain some cycle where all edges are assigned the same color. By construction, every set in $\mathcal{F}$ corresponds to a uniquely colored cycle in $G$. Hence, the contraction operations applied to obtain $G$ cannot create new monochromatic cycles, i.e. every cycle in $G$ which does not correspond to a set from $\mathcal{F}$ must include edges of at least two different colors. Therefore, if $G[V(G) \setminus Y]$ contains some monochromatic cycle then $X$ cannot be a hitting set of $\mathcal{F}$. □

Claim 3. *If $G$ admits an $|\mathcal{F}|$-simfvs of size at most $k$ then $\mathcal{F}$ admits a hitting set of size at most $k$.*

Proof. Let $X = \{v_{i_1}, \ldots, v_{i_k}\}$ be such an $|\mathcal{F}|$-simfvs. First, note that if some vertex in $X$ does not correspond to an element in $\mathcal{U}$, then we can safely replace that vertex with one that does (since any such vertex belongs to exactly one monochromatic cycle). We construct a set $Y = \{u_{i_1}, \ldots, u_{i_k}\}$. If there exists a set $f_i \in \mathcal{F}$ such that $Y \cap f_i = \emptyset$ then, by construction, there exists an $i$-colored cycle $C_i$ in $G$ such that $X \cap V(C_i) = \emptyset$, a contradiction. □

Combining the previous two claims with the fact that our reduction runs in time polynomial in $|\mathcal{U}|$, $|\mathcal{F}|$, and $k$, completes the proof of the theorem. □

Notice that the proof of Theorem 5.1 crucially relies on the fact that each cycle is "uniquely identified" by a separate color. In order to get around this limitation and prove W[1]-hardness for $\alpha \in O(\log n)$ we need, in some sense, to group separate sets of a Hitting Set instance into $O(\log(|\mathcal{U}||\mathcal{F}|))$ families such that sets inside each family are pairwise disjoint. By doing so, we can modify the proof of Theorem 5.1 to identify all sets inside a family using the same color, for a total of $O(\log n)$ colors (instead of $O(n)$). We achieve exactly this in what follows. We refer the reader to the work of Impagliazzo et al. [14, 15] for details on the Exponential Time Hypothesis (ETH). For our purposes, it is enough to view ETH as the following conjecture: There exists a positive real $s$ such that determining whether a 3-CNF boolean formula on $n$ variables and $m$ clauses is satisfiable cannot be accomplished in time $2^{sn}(n + m)^{O(1)}$.

Partitioned Hitting Set (PHS) **Parameter:** $k$
**Input:** A tuple $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k)$, where $\mathcal{F}_i$, $1 \leq i \leq \alpha$, is a collection of subsets of the finite universe $\mathcal{U}$ and $k$ is a positive integer. Moreover, all the sets within a family $\mathcal{F}_i$, $1 \leq i \leq \alpha$, are pairwise disjoint.
**Question:** Is there a subset $X$ of $\mathcal{U}$ of cardinality at most $k$ such that for every $f \in \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha$, $f \cap X$ is nonempty?

Partitioned Subgraph Isomorphism (PSI) **Parameter:** $k = |E(G)|$
**Input:** A graph $H$, a graph $G$ with $V(G) = \{g_1, \ldots, g_\ell\}$, and a coloring function $col : V(H) \longrightarrow [\ell]$.
**Question:** Is there an injection $inj : V(G) \longrightarrow V(H)$ such that for every $i \in [\ell]$, $col(inj(g_i)) = i$ and for every $(g_i, g_j) \in E(G)$, $(inj(g_i), inj(g_j)) \in E(H)$?

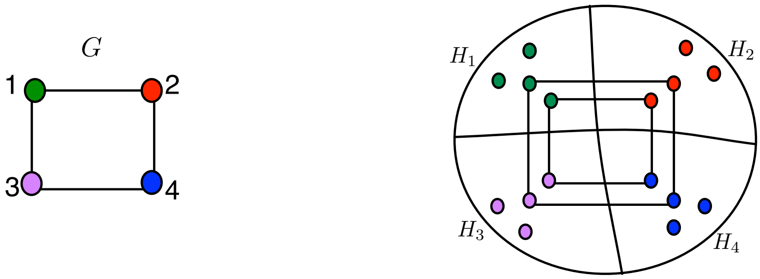

Fig. 5. An instance of the PSI problem.

Theorem 5.2 ([13, 22]). Partitioned Subgraph Isomorphism *parameterized by* $|E(G)|$ *is W[1]-hard, even when the smaller graph G is connected and has maximum degree three. Moreover, the problem cannot be solved in time* $f(k)n^{o(\frac{k}{\log k})}$*, where* $f$ *is an arbitrary function,* $n = |V(H)|$*, and* $k = |E(G)|$*, unless ETH fails.*

We make a few simplifying assumptions: For an instance of Partitioned Subgraph Isomorphism, we let $H_i$ denote the subgraph of $H$ induced on vertices colored $i$. We assume that $|V(H_i)| = 2^t$, for $1 \leq i \leq \ell$ and $t$ some positive integer; adding isolated vertices to each set is enough to guarantee this size constraint. Moreover, we assume that whenever there is no edge $(g_i, g_j) \in E(G)$, then there are no edges between $V(H_i)$ and $V(H_j)$ in $H$ (see Figure 5 for an example of an instance). Note that the PSI problem asks for a "colorful" subgraph of $H$ isomorphic to $G$ such that one vertex from $H_i$ is mapped to the vertex $g_i$, $1 \leq i \leq \ell$. Therefore, it is also safe to assume that $H_i$, $1 \leq i \leq \ell$, is edgeless.

Theorem 5.3. Partitioned Hitting Set *parameterized by solution size is W[1]-hard when* $\alpha \in O(\log(|\mathcal{U}||\mathcal{F}|))$*. Moreover, the problem cannot be solved in time* $f(k)n^{o(\frac{k}{\log k})}$*, where* $f$ *is an arbitrary function,* $n = |\mathcal{U}|$*, and* $k$ *is the required solution size, unless ETH fails.*

Proof. Given an instance $(H, G, col, \ell = |V(G)|, k = |E(G)|)$ of PSI, where $G$ has maximum degree three, we reduce it into an instance $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k' = k + \ell)$ of PHS, where $\alpha = 16 \log 2^t + 1 = 16t + 1$, $\mathcal{F}_i$, $1 \leq i \leq \alpha$, is a collection of subsets of the finite universe $\mathcal{U}$, and all the sets within a family $\mathcal{F}_i$ are pairwise disjoint.

We start by constructing the universe $\mathcal{U}$. For each vertex $h^i_j \in V(H_i)$, $1 \leq i \leq \ell$ and $0 \leq j \leq 2^t - 1$, we create an element $v^i_j$. For each edge $(h^{i_1}_{j_1}, h^{i_2}_{j_2}) \in E(H)$, we create an element $e^{i_1,i_2}_{j_1,j_2}$ where $j_1$ is the index of the vertex in $H_{i_1}$, $j_2$ is the index of the vertex in $H_{i_2}$, $1 \leq i_1, i_2 \leq \ell$, and $0 \leq j_1, j_2 \leq 2^t - 1$. Note that $|\mathcal{U}| = |V(H)| + |E(H)| = \ell 2^t + |E(H)| < 4^t 2\ell^2$.

We now create "selector gadgets" between elements corresponding to vertices and elements corresponding to edges. For every *ordered* pair $(x, y)$, $1 \leq x, y \leq \ell$, such that there exists an edge between $H_x$ and $H_y$ in $H$ (or equivalently there exists an edge $(g_x, g_y)$ in $G$), we create $2t$ sets. We denote half of those sets by $U_{x,y,p}$ and the order half by $D_{x,y,p}$, where $1 \leq p \leq t$. Let $\mathcal{U}_x$ denote the set of all elements corresponding to vertices in $H_x$ and let $\mathcal{U}_{x,y}$ ($x$ and $y$ unordered in $\mathcal{U}_{x,y}$) denote the set of all elements corresponding to edges between vertices in $H_x$ and vertices $H_y$. We let $bit(i)[p]$, $0 \leq i \leq 2^t - 1$ and $1 \leq p \leq t$, be the $p^{th}$ bit in the bit representation of $i$ (where position $p = 1$ holds the most significant bit). For each $v^x_i \in \mathcal{U}_x$ and for all $p$ from 1 to $t$, if $bit(i)[p] = 0$ we add $v^x_i$ to set $D_{x,y,p}$ and we add $v^x_i$ to set $U_{x,y,p}$ otherwise. For each $e^{x,y}_{i,j} \in \mathcal{U}_{x,y}$ and for all $p$ from

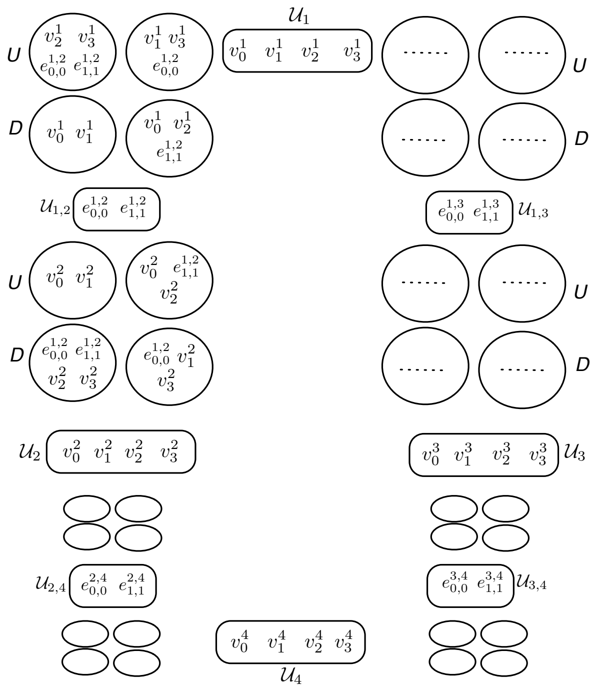
$\mathcal{U}_1$
U
$v_2^1$ $v_3^1$ $e_{0,0}^{1,2}$ $e_{1,1}^{1,2}$
$v_1^1$ $v_3^1$ $e_{0,0}^{1,2}$
$v_0^1$ $v_1^1$ $v_2^1$ $v_3^1$
......
......
U
D
$v_0^1$ $v_1^1$
$v_0^1$ $v_2^1$ $e_{1,1}^{1,2}$
......
......
D
$\mathcal{U}_{1,2}$
$e_{0,0}^{1,2}$ $e_{1,1}^{1,2}$
$e_{0,0}^{1,3}$ $e_{1,1}^{1,3}$
$\mathcal{U}_{1,3}$
U
$v_0^2$ $v_1^2$
$v_0^2$ $e_{1,1}^{1,2}$ $v_2^2$
......
......
U
D
$e_{0,0}^{1,2}$ $e_{1,1}^{1,2}$ $v_2^2$ $v_3^2$
$e_{0,0}^{1,2}$ $v_1^2$ $v_3^2$
......
......
D
$\mathcal{U}_2$
$v_0^2$ $v_1^2$ $v_2^2$ $v_3^2$
$v_0^3$ $v_1^3$ $v_2^3$ $v_3^3$
$\mathcal{U}_3$
$\mathcal{U}_{2,4}$
$e_{0,0}^{2,4}$ $e_{1,1}^{2,4}$
$e_{0,0}^{3,4}$ $e_{1,1}^{3,4}$
$\mathcal{U}_{3,4}$
$v_0^4$ $v_1^4$ $v_2^4$ $v_3^4$
$\mathcal{U}_4$

Fig. 6. Parts of the reduction for the PSI instance from Figure 5. Rounded rectangles represents subsets of the universe and circles (and ellipses) represent sets in the family.

1 to $t$, if $bit(i)[p] = 0$ we add $e_{i,j}^{x,y}$ to set $U_{x,y,p}$ and we add $e_{i,j}^{x,y}$ to set $D_{x,y,p}$ otherwise. Recall that for $e_{i,j}^{x,y}$, $i$ corresponds to the index of element $v_i^x \in \mathcal{U}_x$.

Finally, for each $x$, $1 \leq x \leq \ell$, we add the set $Q_x = \mathcal{U}_x$, and for each (unordered) pair $x$,$y$ such that $(g_x, g_y) \in E(G)$ we add the set $Q_{x,y} = \mathcal{U}_{x,y}$. Put differently, a set $Q_x$ contains all elements corresponding to vertices in $H_x$ and a set $Q_{x,y}$ contains all elements corresponding to edges between $H_x$ and $H_y$. The role of these $\ell + k$ sets is simply to force a solution to pick at least one element from every $\mathcal{U}_x$ and one element from every $\mathcal{U}_{y,z}$, $1 \leq x, y, z \leq \ell$. Note that we have a total of $4t|E(G)| + |E(G)| + \ell < 4t\ell^2 + \ell^2 + \ell$ sets and therefore $16t + 1 \in O(\log(|\mathcal{U}||\mathcal{F}|))$. We set $k' = |V(G)| + |E(G)| = \ell + k$. This completes the construction. An example of the construction for the instance given in Figure 5 is provided in Figure 6.

Claim 4. *In the resulting instance* $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k' = k + \ell)$, $\alpha = 16 \log 2^t + 1 = 16t + 1$.

Proof. First, we note that all sets $Q_x$ and $Q_{y,z}$, $1 \leq x, y, z \leq \ell$, are pairwise disjoint. Hence, we can group all these sets into a single partition. We now prove that $16t$ is enough to partition the remaining sets.

Since $G$ has maximum degree three, we know by Vizing's theorem [28] that $G$ admits a proper 4-edge-coloring, i.e. no two edges incident on the same vertex receive the same color. Let us fix such a 4-edge-coloring and denote it by $\beta : E(G) \longrightarrow \{1, 2, 3, 4\}$. Recall that for every ordered pair $(x, y)$, $1 \leq x, y \leq \ell$, we define two groups of sets $U_{x,y,p}$ and $D_{x,y,p}$, $1 \leq p \leq t$. Given any set $X_{x,y,p}$, $X \in \{U, D\}$, we define the partition to which $X_{x,y,p}$ belongs as $part(X, x, y, p) = (\beta(g_x, g_y), p, \{U, D\}, \{x < y, x > y\})$. In other words, we have a total of $16t$ partitions depending on the color of the edge $(g_x, g_y)$ in $G$, the position $p$, whether $X = U$ or $X = D$, and whether $x < y$ or $x > y$ (recall that we assume $x \neq y$).

Since $\beta$ is a proper 4-coloring of the edges of $G$, we know that if two sets belong to the same partition they must be of the form $X_{x_1,y_1,p}$ and $X_{x_2,y_2,p}$, where $X \in \{U, D\}$, $x_1 \neq x_2$, $y_1 \neq y_2$, $\beta(g_{x_1}, g_{y_1}) = \beta(g_{x_2}, g_{y_2})$, $x_1 < y_1$ ($x_1 > y_1$), and $x_2 < y_2$ ($x_2 > y_2$). It follows from our construction that $X_{x_1,y_1,p} \cap X_{x_2,y_2,p} = \emptyset$; $X_{x_1,y_2,p}$ only contains elements from $\mathcal{U}_{x_1} \cup \mathcal{U}_{x_1,y_1}$, $X_{x_2,y_2,p}$ only contains elements from $\mathcal{U}_{x_2} \cup \mathcal{U}_{x_2,y_2}$, and $(\mathcal{U}_{x_1} \cup \mathcal{U}_{x_1,y_1}) \cap (\mathcal{U}_{x_2} \cup \mathcal{U}_{x_2,y_2})$ is empty. □

Claim 5. *The resulting instance $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k' = k + \ell)$ admits no hitting set of size $k' - 1$.*

Proof. If there exists a hitting set $S$ of size $k' - 1$, then either (1) there exists $\mathcal{U}_x$, $1 \leq x \leq \ell$, such that $S \cap \mathcal{U}_x = \emptyset$ or (2) there exists $\mathcal{U}_{y,z}$, $1 \leq y, z \leq \ell$, such that $S \cap \mathcal{U}_{y,z} = \emptyset$. In case (1), we are left with a set $Q_x$ which is not hit by $S$. Similarly, for case (2), there exists a set $Q_{y,z}$ which is not hit by $S$. In both cases we get a contradiction as we assumed $S$ to be a hitting set, as needed. □

Claim 6. *Any hitting set of size $k'$ of the resulting instance $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k' = k + \ell)$ must pick exactly one element from each set $\mathcal{U}_x$, $1 \leq x \leq \ell$, and exactly one element from each set $\mathcal{U}_{y,z}$, $1 \leq y, z \leq \ell$. Moreover, for every ordered pair $(x, y)$, $1 \leq x, y \leq \ell$, a hitting set of size $k'$ must pick $v^x_i \in \mathcal{U}_x$ and $e^{x,y}_{i,j} \in \mathcal{U}_{x,y}$, $0 \leq i, j \leq 2^t - 1$. In other words, the vertex $h^x_i \in V(H)$ is incident to the edge $(h^x_i, h^y_j) \in E(H)$.*

Proof. The first part of the claim follows from the previous claim combined with the fact that $k' = k + \ell$. For the second part, assume that there exists a hitting set $S$ of size $k'$ such that for some ordered pair, $(x, y)$, $S$ includes $v^x_{i_1} \in \mathcal{U}_x$ and $e^{x,y}_{i_2,j} \in \mathcal{U}_{x,y}$, where $i_1 \neq i_2$. Since $i_1 \neq i_2$, then $bit(i_1)[p] \neq bit(i_2)[p]$ for at least one position $p$. For that position, we know that $v^x_{i_1}$ and $e^{x,y}_{i_2,j}$ must both belong to only one of $U_{x,y,p}$ or $D_{x,y,p}$. Hence, either $U_{x,y,p}$ or $D_{x,y,p}$ is not hit by $v^x_{i_1}$ and $e^{x,y}_{i_2,j}$ when $i_1 \neq i_2$. □

Claim 7. *If $(H, G, col, \ell = |V(G)|, k = |E(G)|)$, where $G$ has maximum degree three, is a yes-instance of* PSI *then $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k' = k + \ell)$ is a yes-instance of* PHS.

Proof. Let $S$, a subgraph of $H$, denote the solution graph and let $V(S) = \{h^1_{i_1}, \ldots, h^\ell_{i_\ell}\}$. We claim that $S' = \{v^1_{i_1}, \ldots, v^\ell_{i_\ell}\} \cup \{e^{x,y}_{j_1,j_2} | (g_x, g_y) \in E(G) \wedge j_1, j_2 \in \{i_1, \ldots, i_\ell\}\}$ is a hitting set of $\mathcal{F}$. That is, the hitting set picks $\ell$ elements corresponding to the $\ell$ vertices in $S$ (or $G$) and $k$ elements corresponding to the $k$ edges in $G$.

Clearly, all sets $Q_x$ and $Q_{y,z}$, $1 \leq x, y, z \leq \ell$, are hit since we pick one element from each. We now show that all sets $U_{x,y,p}$ and $D_{x,y,p}$, $1 \leq x, y \leq \ell$ and $1 \leq p \leq t$, are also hit. Assume, without loss of generality, that for fixed $x$, $y$, and $p$, some set $U_{x,y,p}$ is not hit. Let $v^x_{i_1} \in \mathcal{U}_x$ be the element we picked from $\mathcal{U}_x$ and let $e^{x,y}_{i_2,j}$ be the element we picked from $\mathcal{U}_{x,y}$. If $U_{x,y,p}$ is not hit, it must be the case that $i_1 \neq i_2$ which, by the previous claim, is not possible. □

Claim 8. *If $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k' = k + \ell)$ is a yes-instance of* PHS *then $(H, G, col, \ell = |V(G)|, k = |E(G)|)$ is a yes-instance of* PSI.

Proof. Let $S = \{v^1_{i_1}, \ldots, v^\ell_{i_\ell}\} \cup \{e^{x,y}_{j_1,j_2} | (g_x, g_y) \in E(G) \wedge j_1, j_2 \in \{i_1, \ldots, i_\ell\}\}$ be a hitting set of $\mathcal{F}$. Note that we can safely assume that the hitting set picks such elements since it has to hit all sets $Q_x$ and $Q_{y,z}$, $1 \leq x, y, z \leq \ell$. We claim that the subgraph $S'$ of $H$ with vertex set $V(S') = \{h^1_{i_1}, \ldots, h^\ell_{i_\ell}\}$ is a solution to the PSI instance.

By construction, there is an injection $inj : V(G) \rightarrow V(S')$ such that for every $i \in [\ell]$, $col(inj(g_i)) = i$. In fact, $S'$ contains exactly one vertex for each color $i \in [\ell]$. Assume that there exists an edge $(g_i, g_j) \in E(G)$ such that $(inj(g_i), inj(g_j)) \notin E(S')$. This implies that there exists two vertices $h_i^x, h_j^y \in V(S')$ such that $(h_i^x, h_j^y) \notin E(S')$. But we know that there exists at least one edge, say $(h_{i'}^x, h_{j'}^y)$, between vertices in $H_x$ and vertices in $H_y$ (from our assumptions). Since $i' \neq i$, $j' \neq j$, $v_i^x, v_j^y \in S$, and $e_{i,j}^{x,y} \notin S$, it follows that $S$ cannot be a hitting set of $\mathcal{F}$ as at least one set in $U_{x,y,p} \cup D_{x,y,p}$ and one set in $U_{y,x,p} \cup D_{y,x,p}$ is not hit by $S$, a contradiction. □

This completes the proof of the theorem. □

We are now ready to state the main result of this section. The proof of Theorem 5.4 follows the same steps as the proof of Theorem 5.1 with one exception, i.e we reduce from Partitioned Hitting Set with $\alpha \in O(\log(|\mathcal{U}||\mathcal{F}|))$ and use $O(\log(|\mathcal{U}||\mathcal{F}|))$ colors instead of $|\mathcal{F}|$.

Theorem 5.4. SimFVS *parameterized by solution size is W[1]-hard when $\alpha \in O(\log n)$.*

Proof. Given an instance $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k)$ of PHS, we let $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$ and $\mathcal{F}_i = \{f_1^i, \ldots, f_{|\mathcal{F}_i|}^i\}$, $1 \leq i \leq \alpha$. We assume, without loss of generality, that each element in $\mathcal{U}$ belongs to at least one set in $\mathcal{F}$.

For each $f_j^i \in \mathcal{F}_i$, $1 \leq i \leq \alpha$ and $1 \leq j \leq |\mathcal{F}_i|$, we create a vertex-disjoint cycle $C_j^i$ on $2|\mathcal{U}|$ vertices and assign all its edges color $i$. We let $V(C_j^i) = \{c_1^{i,j}, \ldots, c_{2|\mathcal{U}|}^{i,j}\}$ and we define $\beta(i, j, u_p) = c_{2p-1}^{i,j}$, $1 \leq i \leq \alpha$, $1 \leq j \leq |\mathcal{F}_i|$, and $1 \leq p \leq |\mathcal{U}|$. In other words, every odd-numbered vertex of $C_j^i$ is mapped to an element in $\mathcal{U}$. Now for every element $u_p \in \mathcal{U}$, $1 \leq p \leq |\mathcal{U}|$, we create a vertex $v_p$, we let $\gamma(u_p) = \{c_{2p-1}^{i,j} | 1 \leq i \leq \alpha \wedge 1 \leq j \leq |\mathcal{F}_i| \wedge u_p \in f_j^i\}$, and we add an edge (of some special color, say 0) between $v_p$ and every vertex in $\gamma(u_p)$. To finalize the reduction, we contract all the edges colored 0 to obtain an instance $(G, k)$ of SimFVS. Note that $|V(G)| = |E(G)| = 2|\mathcal{U}||\mathcal{F}|$ and the total number of used colors is $\alpha$. Moreover, after contracting all special edges, $|\gamma(u_p)| = 1$ for all $u_p \in \mathcal{U}$.

Claim 9. *If $\mathcal{F}$ admits a hitting set of size at most $k$ then $G$ admits an $\alpha$-simfvs of size at most $k$.*

Proof. Let $X = \{u_{p_1}, \ldots, u_{p_k}\}$ be such a hitting set. We construct a vertex set $Y = \{\gamma(u_{p_1}), \ldots, \gamma(u_{p_k})\}$. If $Y$ is not an $\alpha$-simfvs of $G$ then $G[V(G) \setminus Y]$ must contain some monochromatic cycle. By construction, only sets from the same family $\mathcal{F}_i$, $1 \leq i \leq \alpha$, correspond to cycles assigned the same color in $G$. But since we started with an instance of PHS, no two such sets intersect. Hence, the contraction operations applied to obtain $G$ cannot create new monochromatic cycles. Therefore, if $G[V(G) \setminus Y]$ contains some monochromatic cycle then $X$ cannot be a hitting set of $\mathcal{F}$. □

Claim 10. *If $G$ admits an $\alpha$-simfvs of size at most $k$ then $\mathcal{F}$ admits a hitting set of size at most $k$.*

Proof. Let $X = \{v_{p_1}, \ldots, v_{p_k}\}$ be such an $\alpha$-simfvs. First, note that if some vertex in $X$ does not correspond to an element in $\mathcal{U}$, then we can safely replace that vertex with one that does (since any such vertex belongs to exactly one monochromatic cycle). We construct a set $Y = \{u_{p_1}, \ldots, u_{p_k}\}$. If there exists a set $f_j^i \in \mathcal{F}_i$ such that $Y \cap f_j^i = \emptyset$ then, by construction, there exists an $i$-colored cycle $C_i$ in $G$ such that $X \cap V(C_i) = \emptyset$, a contradiction. □

Combining the previous two claims with the fact that our reduction runs in time polynomial in $|\mathcal{U}|$, $|\mathcal{F}|$, and $k$, completes the proof of the theorem. □

## 6 CONCLUSION

We have showed that SimFVS parameterized by solution size $k$ is fixed-parameter tractable and can be solved by an algorithm running in $O^\star(23^{\alpha k})$ time, for any constant $\alpha$. For the special case of $\alpha = 2$, we gave a faster $O^\star(81^k)$ time algorithm which follows from the observation that the base case of the general algorithm can be solved in polynomial time when $\alpha = 2$. Moreover, for constant $\alpha$, we presented a kernel for the problem with $O(\alpha k^{3(\alpha+1)})$ vertices.

It is interesting to note that our algorithm implies that SimFVS can be solved in $(2^{O(\alpha)})^k n^{O(1)}$ time. However, we have also seen that SimFVS becomes W[1]-hard when $\alpha \in O(\log n)$. This implies that (under plausible complexity assumptions) an algorithm running in $(2^{o(\alpha)})^k n^{O(1)}$ time cannot exist. In other words, the running time cannot be subexponential in either $k$ or $\alpha$.

As mentioned by Cai and Ye [3], we believe that studying generalizations of other classical problems to edge-colored graphs is well motivated and might lead to interesting new insights about combinatorial and structural properties of such problems. Some of the potential candidates are Vertex Planarization, Odd Cycle Transversal, Interval Vertex Deletion, Chordal Vertex Deletion, Planar $\mathcal{F}$-Deletion, and, more generally, Simultaneous $\mathcal{F}$-Deletion.

## REFERENCES

[1] Vineet Bafna, Piotr Berman, and Toshihiro Fujito. 1999. A 2-Approximation Algorithm for the Undirected Feedback Vertex Set Problem. *SIAM J. Discrete Math.* 12, 3 (1999), 289–297. https://doi.org/10.1137/S0895480196305124

[2] Jørgen Bang-Jensen and Gregory Gutin. 1997. Alternating cycles and paths in edge-coloured multigraphs: A survey. *Discrete Mathematics* 165-166 (1997), 39–60. https://doi.org/10.1016/S0012-365X(96)00160-4

[3] Leizhen Cai and Junjie Ye. 2014. Dual Connectedness of Edge-Bicolored Graphs and Beyond. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II.* 141–152. https://doi.org/10.1007/978-3-662-44465-8_13

[4] Yixin Cao and Dániel Marx. 2015. Interval Deletion Is Fixed-Parameter Tractable. *ACM Trans. Algorithms* 11, 3 (2015), 21:1–21:35. https://doi.org/10.1145/2629595

[5] Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. 2008. Improved algorithms for feedback vertex set problems. *J. Comput. Syst. Sci.* 74, 7 (2008), 1188–1198. https://doi.org/10.1016/j.jcss.2008.05.002

[6] Jianer Chen, Iyad A. Kanj, and Ge Xia. 2010. Improved upper bounds for vertex cover. *Theor. Comput. Sci.* 411, 40-42 (2010), 3736–3756. https://doi.org/10.1016/j.tcs.2010.06.026

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

[8] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. 2011. Solving Connectivity Problems Parameterized by Treewidth in Single Exponential Time. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011.* 150–159. https://doi.org/10.1109/FOCS.2011.23

[9] Reinhard Diestel. 2012. *Graph Theory, 4th Edition.* Graduate texts in mathematics, Vol. 173. Springer.

[10] Rod G. Downey and Michael R. Fellows. 1997. *Parameterized complexity.* Springer-Verlag.

[11] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. 2012. Planar F-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012.* 470–479. https://doi.org/10.1109/FOCS.2012.62

[12] M. R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman.

[13] Martin Grohe and Dániel Marx. 2009. On tree width, bramble size, and expansion. *J. Comb. Theory, Ser. B* 99, 1 (2009), 218–228. https://doi.org/10.1016/j.jctb.2008.06.004

[14] Russell Impagliazzo and Ramamohan Paturi. 2001. On the Complexity of k-SAT. *J. Comput. Syst. Sci.* 62, 2 (2001), 367–375. https://doi.org/10.1006/jcss.2000.1727

[15] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. 2001. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.* 63, 4 (2001), 512–530. https://doi.org/10.1006/jcss.2001.1774

[16] Mikio Kano and Xueliang Li. 2008. Monochromatic and Heterochromatic Subgraphs in Edge-Colored Graphs - A Survey. *Graphs and Combinatorics* 24, 4 (2008), 237–263. https://doi.org/10.1007/s00373-008-0789-5

[17] Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. 2016. Linear Kernels and Single-Exponential Algorithms Via Protrusion Decompositions. *ACM Trans. Algorithms* 12, 2 (2016), 21:1–21:41. https://doi.org/10.1145/2797140

[18] Tomasz Kociumaka and Marcin Pilipczuk. 2014. Faster deterministic Feedback Vertex Set. *Inf. Process. Lett.* 114, 10 (2014), 556–560. https://doi.org/10.1016/j.ipl.2014.05.001

[19] Stefan Kratsch and Magnus Wahlström. 2012. Representative Sets and Irrelevant Vertices: New Tools for Kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23,*

*2012*. 450–459. https://doi.org/10.1109/FOCS.2012.46

[20] Daniel Lokshtanov. 2008. Wheel-Free Deletion Is W[2]-Hard. In *Parameterized and Exact Computation, Third International Workshop, IWPEC 2008, Victoria, Canada, May 14-16, 2008. Proceedings*. 141–147. https://doi.org/10.1007/978-3-540-79723-4_14

[21] Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. 2014. Faster Parameterized Algorithms Using Linear Programming. *ACM Trans. Algorithms* 11, 2 (2014), 15:1–15:31. https://doi.org/10.1145/2566616

[22] Dániel Marx. 2010. Can You Beat Treewidth? *Theory of Computing* 6, 1 (2010), 85–112. https://doi.org/10.4086/toc.2010.v006a005

[23] Dániel Marx. 2010. Chordal Deletion is Fixed-Parameter Tractable. *Algorithmica* 57, 4 (2010), 747–768. https://doi.org/10.1007/s00453-008-9233-8

[24] Dániel Marx and Ildikó Schlotter. 2012. Obtaining a Planar Graph by Vertex Deletion. *Algorithmica* 62, 3-4 (2012), 807–822. https://doi.org/10.1007/s00453-010-9484-z

[25] Pranabendu Misra, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. 2012. Parameterized Algorithms for Even Cycle Transversal. In *Graph-Theoretic Concepts in Computer Science - 38th International Workshop, WG 2012, Jerusalem, Israel, June 26-28, 2012, Revised Selcted Papers*. 172–183. https://doi.org/10.1007/978-3-642-34611-8_19

[26] Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. 2004. Finding odd cycle transversals. *Oper. Res. Lett.* 32, 4 (2004), 299–301. https://doi.org/10.1016/j.orl.2003.10.009

[27] Stéphan Thomassé. 2010. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms* 6, 2 (2010), 32:1–32:8. https://doi.org/10.1145/1721837.1721848

[28] V. G. Vizing. 1964. On an estimate of the chromatic class of a $p$-graph. *Diskret. Analiz No.* 3 (1964), 25–30.

[29] Mingyu Xiao and Hiroshi Nagamochi. 2015. An improved exact algorithm for undirected feedback vertex set. *J. Comb. Optim.* 30, 2 (2015), 214–241. https://doi.org/10.1007/s10878-014-9737-x

[30] Junjie Ye. 2015. A Note on Finding Dual Feedback Vertex Set. *CoRR* abs/1510.00773 (2015). http://arxiv.org/abs/1510.00773

Received February 2007; revised March 2009; accepted June 2009