Simultaneous Feedback Vertex Set: A Parameterized Perspective*

Akanksha Agrawal[1], Daniel Lokshtanov[2], Amer E. Mouawad[3], and Saket Saurabh[4]

1 **Department of Informatics, University of Bergen, Bergen, Norway**
akanksha.agrawal@uib.no
2 **Department of Informatics, University of Bergen, Bergen, Norway**
daniel.lokshtanov@uib.no
3 **Department of Informatics, University of Bergen, Bergen, Norway**
a.mouawad@uib.no
4 **Department of Informatics, University of Bergen, Bergen, Norway; and Institute of Mathematical Sciences, Chennai, India**
saket@imsc.res.in

Abstract

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

1998 ACM Subject Classification G.2.2 Graph Algorithms, I.1.2 Analysis of Algorithms

Keywords and phrases parameterized complexity, feedback vertex set, kernel, edge-colored graphs

Digital Object Identifier 10.4230/LIPIcs.STACS.2016.7

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 does not contain a graph from

* The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no. 306992.

$\mathcal{F}$? 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. 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 [22], 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 $G = (V, \cup_{i=1}^{\alpha} E_i)$, where the edge set of G is partitioned into α color classes, is called an *α-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 sub-graphs, heterochromatic subgraphs, and partitions". A natural extension of the $\mathcal{F}$-DELETION problem to edge-colored graphs is the α-SIMULTANEOUS $\mathcal{F}$-DELETION problem. In the latter problem, we are given an α-edge-colored graph G and the goal is to find a set S of at most k vertices such that each graph $G_i \setminus S$, 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_{red} and G_{blue} are trees. On the positive side, they prove that DUALLY CONNECTED INDUCED SUBGRAPH is solvable in time polynomial in the input size when G is a complete graph. One of the open problems they state is to determine the parameterized complexity of α-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 α-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 $G \setminus S$ is a forest. For an α-colored graph G, an *α-simultaneous feedback vertex set* (or *α-simfvs* for short) is a subset S of vertices such that $G_i \setminus S$ is a forest for each $1 \leq i \leq \alpha$. The α-SIMULTANEOUS FEEDBACK VERTEX SET is stated formally as follows.

α-SIMULTANEOUS FEEDBACK VERTEX SET (α-SIMFVS) **Parameter:** k
Input: (G, k), where G is an undirected α-colored graph and k is a positive integer
Question: Is there a subset $S \subseteq V(G)$ of size at most k such that for $1 \leq i \leq \alpha$, $G_i \setminus S$ 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 $\mathcal{O}^\star(1.736^n)$ time [28], a deterministic algorithm running in $\mathcal{O}^\star(3.619^k)$ time [18], a randomized algorithm running in $\mathcal{O}^\star(3^k)$ time [8], and a kernel on $\mathcal{O}(k^2)$ vertices [27] (see Section 2 for definitions). We use the $\mathcal{O}^\star$ notation to describe the running times of our algorithms. A running time $\mathcal{O}^\star(f(k))$ means that the running time is upper bounded by $f(k)n^{\mathcal{O}(1)}$, where n is the input size. That is, the $\mathcal{O}^\star$ notation suppresses polynomial factors in the running-time expression.

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

- An FPT algorithm running in $\mathcal{O}^\star(23^{\alpha k})$ time. For the special case of $\alpha = 2$, we give a faster algorithm running in $\mathcal{O}^\star(81^k)$ time.
- For constant α, we obtain a kernel with $\mathcal{O}(\alpha k^{3(\alpha+1)})$ vertices.
- The running time of our algorithm implies that α-SimFVS is FPT even when $\alpha \in o(\log n)$. We complement this positive result by showing that for $\alpha \in \mathcal{O}(\log n)$, where n is the number of vertices in the input graph, α-SimFVS 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 α-SimFVS problem, 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_{red} nor G_{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 α-SimFVS problem, even for $\alpha = 2$. For this reason, we also believe that studying α-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 α-SimFVS problem, parameterized by solution size k, in $\mathcal{O}^\star(23^{\alpha k})$ 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 α-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. 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 α-simultaneous feedback 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, $\mathcal{O}^\star(81^k)$ 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 α. Our kernel has $\mathcal{O}(\alpha k^{3(\alpha+1)})$ 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 a feedback vertex set in G_i and study how such paths interact with each other. Using 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 α and both the size of our kernel and the running time of our algorithm in Section 5. We show that even for $\alpha \in \mathcal{O}(\log n)$, where n is the number of vertices in the input graph, α-SimFVS becomes W[1]-hard. We show hardness via a new problem of independent interest which we denote by α-Partitioned Hitting Set. The input to this problem consists of 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 $\mathcal{U}$, k is a positive integer, and all the sets within a family $\mathcal{F}_i$, $1 \leq i \leq \alpha$, are pairwise disjoint. The goal is to determine whether there exists 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. We show that $\mathcal{O}(\log |\mathcal{U}||\mathcal{F}|)$-Partitioned Hitting Set is W[1]-hard via a reduction from Partitioned Subgraph Isomorphism and we show that $\mathcal{O}(\log n)$-SimFVS is W[1]-hard via a reduction from $\mathcal{O}(\log |\mathcal{U}||\mathcal{F}|)$-Partitioned Hitting Set. Along the way, we also show, using a somewhat simpler reduction from Hitting Set, that $\mathcal{O}(n)$-SimFVS is W[2]-hard.

Most of the technical details and proofs have been omitted from this extended abstract.

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.

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 multi-edges. 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 $G \setminus W$ 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 $(v_i, v_{(i+1) \mod k})$ is an edge for all $0 \leq i \leq k$. 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 2 to the degree of v.

An edge α-colored graph is a graph $G = (V, \cup_{i=1}^{\alpha} E_i)$. We call G_i the color i (or i-color) graph of G, where $G_i = (V, E_i)$. For notational convenience we sometimes denote an α-colored graph as $G = (V, E_1, E_2, ..., E_\alpha)$. For an α-colored graph G, the *total degree* of a vertex v is $\sum_{i=1}^{\alpha} d_{G_i}(v)$. 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 α-colored graph

$G = (V, E_1, E_2, \cdots, E_\alpha)$ is an *α-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.

3 FPT Algorithm for α-Simultaneous Feedback Vertex Set

We give an algorithm for the α-SimFVS problem using the method of iterative compression [26, 7]. We only describe the algorithm for the disjoint version of the problem. The existence of an algorithm running in $c^k \cdot n^{\mathcal{O}(1)}$ time for the disjoint variant implies that α-SimFVS can be solved in time $(1+c)^k \cdot n^{\mathcal{O}(1)}$ [7]. In the Disjoint α-SimFVS problem, we are given an α-colored graph $G = (V, E_1, E_2, \ldots, E_\alpha)$, an integer k, and an α-simfvs W in G of size $k+1$. The objective is to find an α-simfvs $X \subseteq V(G) \setminus W$ of size at most k, or correctly conclude the non-existence of such an α-simfvs.

3.1 Algorithm for Disjoint α-SimFVS

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

- Reduction α-SimFVS.R1. Delete isolated vertices as they do not participate in any cycle.
- Reduction α-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 α-SimFVS.R3. If there is a vertex $v \in V(G)$ with exactly two neighbors u, w (the total degree of v is 2), 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 α-SimFVS.R2 has been applied, both edges (v, u) and (v, w) must be of the same color.
- Reduction α-SimFVS.R4. If for some i, $i \in \{1, 2, \ldots, \alpha\}$, there is an edge of multiplicity larger than 2 in E_i, reduce its multiplicity to 2.
- Reduction α-SimFVS.R5. If there is a vertex v with a self loop, then add v to the solution set X, delete v (and all edges incident on v) from the graph and decrease k by 1.

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 3.

Algorithm. We give an algorithm for the decision version of the Disjoint α-SimFVS problem, which only verifies whether a solution exists or not. Such an algorithm can be easily modified to find an actual solution X. We follow a branching strategy with a nontrivial measure function. Let (G, W, k) be an instance of the problem, where G is an α-colored graph. If $G[W]$ is not an α-forest then we can safely return that (G, W, k) is a no-instance. Hence, we assume that $G[W]$ is an α-forest in what follows. Whenever any of our reduction rules α-SimFVS.R1 to α-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 again a no-instance.

Recall that initially F is an α-forest, as W is an α-simfvs. We will consider each forest F_i, for $i \in \{1, 2, \dots, \alpha\}$, separately (where F_i is the color i graph of the α-forest F). For $i \in \{1, 2, \dots, \alpha\}$, we let $W_i = (W, E_i(G[W]))$ and η_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 $\mathcal{C}_i$, for each i, and we let $\mathfrak{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_\alpha\}$. Initially, $\mathcal{C}_i = \emptyset$, for each $i \in \{1, 2, \dots, \alpha\}$. Each cycle that we add to $\mathcal{C}_i$ will be vertex disjoint from previously added cycles. Hence, if at any point $|\mathcal{C}_i| > k$, 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 any cycles containing v. For the running time analysis of our algorithm we will consider the following measure:

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

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 W_i.
- The subtree T_v^i rooted at v contains two vertices u and w which have at least one color i neighbor in W_i (v can be equal to u or w).

▶ **Lemma 1.** *For $i \in \{1, 2, \dots, \alpha\}$, let v_c be a cordate vertex of highest index in some tree of the forest F_i and let $\mathcal{T}_{v_c}$ denote the subtree rooted at v_c. Furthermore, let u_c be one of the vertices in $\mathcal{T}_{v_c}$ such that u_c has a neighbor in W_i. Then, in the path $P = u_c, x_1, \dots, x_t, v_c$ (t could be equal to zero) between u_c and v_c the vertices $x_1, \dots, x_t$ 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. We branch as follows:

- v_c belongs to the solution. We delete v_c from G and decrease k by 1. In this branch μ decreases by α.

  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 1. 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 $G_i \setminus W$, which is itself a component in $G_i \setminus W$. 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 μ 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 $G_i \setminus W$ as a component. We add C to $\mathcal{C}_i$ and delete edges in C from $G_i \setminus W$. In this branch $|\mathcal{C}_i|$ increases by 1, so the measure μ 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:

- v_c belongs to the solution. We delete v_c from G and decrease k by 1. In this branch μ decreases by α.
- 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 $G_i \setminus W$ as a component. As in Case 1, we add C to $\mathcal{C}_i$ and delete edges in C from $G_i \setminus W$. $|\mathcal{C}_i|$ increases by 1, so the measure μ 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 $G_i \setminus W$ as a component. We add C to $\mathcal{C}_i$ and delete edges in C from $G_i \setminus W$. In this branch $|\mathcal{C}_i|$ increases by 1, so the measure μ 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 $(G \setminus P_v, W \cup P_v, k)$. The number of components in W_i decreases and we get a drop of 1 in η_i, so μ 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 $C \in \cup_{i=1}^{\alpha} \mathcal{C}_i$ and let $\mathcal{U} = \cup_{i=1}^{\alpha}(\cup_{C \in \mathcal{C}_i} V(C))$. 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 2.** Disjoint α-SimFVS *is solvable in time* $\mathcal{O}^\star(22^{\alpha k})$.

▶ **Theorem 3.** α-Simultaneous Feedback Vertex Set *is solvable in time* $\mathcal{O}^\star(23^{\alpha k})$.

4 Polynomial Kernel for α-Simultaneous Feedback Vertex Set

In this section we give a kernel with $\mathcal{O}(\alpha k^{3(\alpha+1)})$ vertices for α-SimFVS. Let (G, k) be an instance of α-SimFVS, where G is an α-colored graph and k is a positive integer. We assume that reduction rules α-SimFVS.R1 to α-SimFVS.R5 have been exhaustively applied. The kernelization algorithm then proceeds in two stages. In stage one, we bound the maximum 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 G be a bipartite graph with vertex bipartition (A, B). A set of edges $M \subseteq E(G)$ 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) M saturates exactly $q|A|$ vertices in B.

▶ **Lemma 4** (Expansion Lemma [7]). *Let q be a positive integer and G 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 G.

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 $v \in V(G)$. We make use of the following lemma which easily follows by adapting Lemma 6.8 from the work of Misra et al. [25].

▶ **Lemma 5.** *Let G be an undirected α-colored multi-graph and x be a vertex without a self loop in G_i, for $i \in \{1, 2, \ldots, \alpha\}$. Then in polynomial time we can either decide that (G, k) is a no-instance of α-*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 α-SimFVS.R1 to α-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 self loop. Now consider a vertex v whose degree in G_i is more than $3k(k+4)$. By Proposition 5, we know that one of three cases must apply:
1. (G, k) is a no-instance of α-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 $G_i \setminus H_v$ 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 α-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 X and decrease k by 1. The resulting instance is $(G \setminus \{v\}, k-1)$.

When in case (3), we bound the degree of v as follows. Consider the graph $G'_i = G_i \setminus (H_v \cup \{v\} \cup V_0^i)$, where V_0^i is the set of degree 0 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 α-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 α-SimFVS.**R7.** Let v be a vertex of degree at least $3k(k+4)$ in G_i, for $i \in \{1, 2, \ldots, \alpha\}$, 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 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).

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

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 α-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 $S = \cup_{i=1}^{\alpha} S_i$. Note that S is an α-simfvs in G and has size at most $2\alpha|S_{OPT}|$, where $|S_{OPT}|$ is an optimal α-simfvs in G. Let $F_i = G_i \setminus S_i$. 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 by the vertices in T^i_2 is a set of vertex-disjoint paths. We say a set of distinct vertices $P = \{v_1, \ldots, v_\ell\}$ in T^i_2 forms a *degree-two path* if (v_j, v_{j+1}) is an edge, for all $1 \leq j \leq \ell$, and all vertices $\{v_1, \ldots, v_\ell\}$ have degree exactly two in G_i. We say P is a *maximal degree-two path* if no proper superset of P also forms a degree-two path.

We enumerate all the maximal degree-two paths in $G_i \setminus S_i$, for $i \in \{1, 2, \ldots, \alpha\}$. Let this set of paths in $G_i \setminus S_i$ 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 $G_i \setminus S_i$. We introduce a special symbol ϕ and add ϕ 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 τ, 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 *Intercept*(P^i_j, τ) to be the set of vertices in path P^i_j which are present in all the paths in the tuple (of course a ϕ entry does not contribute to this set). Formally, *Intercept*$(P^i_j, \tau) = \emptyset$ if $P^i_j \notin \tau$ otherwise *Intercept*$(P^i_j, \tau) = \{v \in V(P^i_j) |$ for all $1 \leq t \leq \alpha$, if $t(\tau) \neq \phi$ then $v \in V(t(\tau))\}$.

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 $u \in$ *Intercept*(P^i_j, τ) by creating a separate copy of u for each path.

Formally, for a path $P_j^i \in \mathcal{P}_i$, $\tau \in \mathfrak{S}$, and a vertex $u \in$*Intercept*(P_j^i, τ), the *Unravel*(P_j^i, τ, 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 $t(\tau)$. 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.

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

▶ **Theorem 6.** α-SimFVS *admits a kernel on* $\mathcal{O}(\alpha k^{3(\alpha+1)})$ *vertices.*

Proof. Consider an α-colored graph G on which reduction rules α-SimFVS.R1 to α-SimFVS.R8 have been exhaustively applied. 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. Hence, 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 $S = \cup_{i=1}^{\alpha} S_i$ is a 2α-approximate α-simfvs in G. Let $F_i = G_i \setminus S_i$. 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 $\mathcal{O}(k^2)$ in G_i, for $i \in \{1, 2, \ldots, \alpha\}$. In particular, the degree of each $s \in S$ is bounded by $\mathcal{O}(k^2)$ 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 $|T^i_{\leq 1}| \in \mathcal{O}(k^3)$.

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, $|T^i_{\geq 3}| \in \mathcal{O}(k^3)$. Also, in a tree, the number of maximal degree-two paths is bounded by $t + l$. Consequently, the number of degree-two paths in $G_i \setminus S_i$ is in $\mathcal{O}(k^3)$. Moreover, no two maximal degree-two paths in a tree intersect.

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

5 Hardness Results

In this section we show that $\mathcal{O}(\log n)$-SimFVS, where n is the number of vertices in the input graph, is W[1]-hard. We give a reduction from a special version of the Hitting Set (HS) problem, which we denote by α-Partitioned Hitting Set (α-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 α-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 $\mathcal{O}(n)$-SimFVS parameterized by solution size is W[2]-hard . 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 7.** $\mathcal{O}(n)$-SimFVS *parameterized by solution size is W[2]-hard.*

Notice that if we assume that $|\mathcal{U}|$ and $|\mathcal{F}|$ are linearly dependent, then Theorem 7 in fact shows that $\mathcal{O}(\sqrt{n})$-SimFVS is W[2]-hard. However, the construction of Theorem 7 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 of $\mathcal{O}(\log n)$-SimFVS we need, in some sense, to group separate sets of a Hitting Set instance into $\mathcal{O}(\log(|\mathcal{U}||\mathcal{F}|))$ families such that sets inside each family are pairwise disjoint. By doing so, we can modify the reduction of Theorem 7 to identify all sets inside a family using the same color, for a total of $\mathcal{O}(\log n)$ colors (instead of $\mathcal{O}(n)$ or $\mathcal{O}(\sqrt{n})$). 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).

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

▶ **Theorem 8** ([13, 23])**.** Partitioned Subgraph Isomorphism *parameterized by* $|E(G)|$ *is W[1]-hard, even when the maximum degree of the smaller graph G is 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.*

▶ **Theorem 9.** $\mathcal{O}(\log(|\mathcal{U}||\mathcal{F}|))$-Partitioned Hitting Set *parameterized by solution size is W[1]-hard. 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.*

We are now ready to state the main result of this section. The proof of Theorem 10 follows the same steps as the proof of Theorem 7 with one exception, i.e we reduce from $\mathcal{O}(\log(|\mathcal{U}||\mathcal{F}|))$-Partitioned Hitting Set and use $\mathcal{O}(\log(|\mathcal{U}||\mathcal{F}|))$ colors instead of $|\mathcal{F}|$.

▶ **Theorem 10.** $\mathcal{O}(\log n)$-SimFVS *parameterized by solution size is W[1]-hard.*

Proof. Given an instance $(\mathcal{U}, \mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_\alpha, k)$ of α-PHS, we let $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$ and $\mathcal{F}_i = \{f^i_1, \ldots, f^i_{|\mathcal{F}_i|}\}$, $1 \leq i \leq \alpha$. We assume, without loss of generality, that each element in $\mathcal{U}$ belongs to at least one set in $\mathcal{F}$.

For each $f^i_j \in \mathcal{F}_i$, $1 \leq i \leq \alpha$ and $1 \leq j \leq |\mathcal{F}_i|$, we create a vertex-disjoint cycle C^i_j on $2|\mathcal{U}|$ vertices and assign all its edges color i. We let $V(C^i_j) = \{c^{i,j}_1, \ldots, c^{i,j}_{2|\mathcal{U}|}\}$ and we define $\beta(i, j, u_p) = c^{i,j}_{2p-1}$, $1 \leq i \leq \alpha$, $1 \leq j \leq |\mathcal{F}_i|$, and $1 \leq p \leq |\mathcal{U}|$. In other words, every odd-numbered vertex of C^i_j 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 $\gamma(u_p) = \{c^{i,j}_{2p-1} | 1 \leq i \leq \alpha \wedge 1 \leq j \leq |\mathcal{F}_i| \wedge u_p \in f^i_j\}$, 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 $\mathcal{O}(\log n)$-SimFVS. Note that $|V(G)| = |E(G)| = 2|\mathcal{U}||\mathcal{F}|$ and the total number of used colors is α. Moreover, after contracting all special edges, $|\gamma(u_p)| = 1$ for all $u_p \in \mathcal{U}$.

▶ **Claim 1.** *If $\mathcal{F}$ admits a hitting set of size at most k then G admits an α-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 α-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 α-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 2.** *If G admits an α-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 α-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 $f^i_j \in \mathcal{F}_i$ such that $Y \cap f^i_j = \emptyset$ 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 α-SimFVS parameterized by solution size k is fixed-parameter tractable and can be solved by an algorithm running in $\mathcal{O}^\star(23^{\alpha k})$ time, for any constant α. For the special case of $\alpha = 2$, we gave a faster $\mathcal{O}^\star(81^k)$ 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 α, we presented a kernel for the problem with $\mathcal{O}(\alpha k^{3(\alpha+1)})$ vertices.

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

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, α-Simultaneous $\mathcal{F}$-Deletion.

References

1 Vineet Bafna, Piotr Berman, and Toshihiro Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM J. Discret. Math.*, 12(3):289–297, September 1999. URL: `http://dx.doi.org/10.1137/S0895480196305124`, `doi:10.1137/S0895480196305124`.

2 Jorgen Bang-Jensen and Gregory Gutin. Alternating cycles and paths in edge-coloured multigraphs: A surve. *Discrete Mathematics*, 165–166:39–60, 1997. Graphs and Combinatorics.

3 Leizhen Cai and Junjie Ye. Dual connectedness of edge-bicolored graphs and beyond. In *Mathematical Foundations of Computer Science 2014*, volume 8635 of *Lecture Notes in Computer Science*, pages 141–152. Springer Berlin Heidelberg, 2014. URL: `http://dx.doi.org/10.1007/978-3-662-44465-8_13`.

4 Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Trans. Algorithms*, 11(3):21:1–21:35, January 2015. URL: `http://doi.acm.org/10.1145/2629595`, `doi:10.1145/2629595`.

5 Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for feedback vertex set problems. *Journal of Computer and System Sciences*, 74(7):1188–1198, 2008. URL: `http://www.sciencedirect.com/science/article/pii/S0022000008000500`, `doi:http://dx.doi.org/10.1016/j.jcss.2008.05.002`.

6 Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, September 2010. URL: `http://dx.doi.org/10.1016/j.tcs.2010.06.026`, `doi: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. *Parameterized Algorithms*. Springer, 2015. URL: `http://dx.doi.org/10.1007/978-3-319-21275-3`, `doi:10.1007/978-3-319-21275-3`.

8 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Joham M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS'11, pages 150–159, Washington, DC, USA, 2011. IEEE Computer Society. URL: `http://dx.doi.org/10.1109/FOCS.2011.23`, `doi:10.1109/FOCS.2011.23`.

9 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

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

11 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-Deletion: Approximation, kernelization and optimal FPT algorithms. In *FOCS*, pages 470–479. IEEE Computer Society, 2012. URL: `http://dblp.uni-trier.de/db/conf/focs/focs2012.html#FominLMS12`.

12 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

13 Martin Grohe and Dániel Marx. On tree width, bramble size, and expansion. *Journal of Combinatorial Theory, Series B*, 99(1):218–228, 2009. URL: `http://www.sciencedirect.com/science/article/pii/S0095895608000683`, `doi:http://dx.doi.org/10.1016/j.jctb.2008.06.004`.

14 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. URL: `http://www.`

sciencedirect.com/science/article/pii/S0022000000917276, doi:http://dx.doi.org/10.1006/jcss.2000.1727.

15 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. URL: http://www.sciencedirect.com/science/article/pii/S002200000191774X, doi:http://dx.doi.org/10.1006/jcss.2001.1774.

16 Mikio Kano and Xueliang Li. Monochromatic and heterochromatic subgraphs in edge-colored graphs – a survey. *Graphs and Combinatorics*, 24(4):237–263, 2008.

17 Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. In *Automata, Languages, and Programming*, volume 7965 of *Lecture Notes in Computer Science*, pages 613–624. Springer Berlin Heidelberg, 2013. URL: http://dx.doi.org/10.1007/978-3-642-39206-1_52.

18 Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. *Information Processing Letters*, 114(10):556–560, 2014. URL: http://www.sciencedirect.com/science/article/pii/S0020019014000969, doi:http://dx.doi.org/10.1016/j.ipl.2014.05.001.

19 Stefan Kratsch and Magnus Wahlstrom. Representative sets and irrelevant vertices: New tools for kernelization. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 0:450–459, 2012. doi:http://doi.ieeecomputersociety.org/10.1109/FOCS.2012.46.

20 Daniel Lokshtanov. Wheel-free deletion is W[2]-Hard. In Martin Grohe and Rolf Niedermeier, editors, *Parameterized and Exact Computation*, volume 5018 of *Lecture Notes in Computer Science*, pages 141–147. Springer Berlin Heidelberg, 2008. URL: http://dx.doi.org/10.1007/978-3-540-79723-4_14, doi:10.1007/978-3-540-79723-4_14.

21 Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, October 2014. URL: http://doi.acm.org/10.1145/2566616, doi:10.1145/2566616.

22 Dániel Marx. Chordal deletion is fixed-parameter tractable. In FedorV. Fomin, editor, *Graph-Theoretic Concepts in Computer Science*, volume 4271 of *Lecture Notes in Computer Science*, pages 37–48. Springer Berlin Heidelberg, 2006. URL: http://dx.doi.org/10.1007/11917496_4, doi:10.1007/11917496_4.

23 Dániel Marx. Can you beat treewidth? In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS'07, pages 169–179, Washington, DC, USA, 2007. IEEE Computer Society. URL: http://dx.doi.org/10.1109/FOCS.2007.18, doi:10.1109/FOCS.2007.18.

24 Dániel Marx and Ildiko Schlotter. Obtaining a planar graph by vertex deletion. In *Graph-Theoretic Concepts in Computer Science*, volume 4769 of *Lecture Notes in Computer Science*, pages 292–303. Springer Berlin Heidelberg, 2007. URL: http://dx.doi.org/10.1007/978-3-540-74839-7_28.

25 Pranabendu Misra, Venkatesh Raman, M.S. Ramanujan, and Saket Saurabh. Parameterized algorithms for even cycle transversal. In MartinCharles Golumbic, Michal Stern, Avivit Levy, and Gila Morgenstern, editors, *Graph-Theoretic Concepts in Computer Science*, volume 7551 of *Lecture Notes in Computer Science*, pages 172–183. Springer Berlin Heidelberg, 2012. URL: http://dx.doi.org/10.1007/978-3-642-34611-8_19, doi:10.1007/978-3-642-34611-8_19.

26 Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004. URL: http://www.sciencedirect.com/science/

article/pii/S0167637703001482, doi:http://dx.doi.org/10.1016/j.orl.2003.10.009.

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

28 Mingyu Xiao and Hiroshi Nagamochi. An improved exact algorithm for undirected feedback vertex set. In Peter Widmayer, Yinfeng Xu, and Binhai Zhu, editors, *Combinatorial Optimization and Applications*, volume 8287 of *Lecture Notes in Computer Science*, pages 153–164. Springer International Publishing, 2013. URL: http://dx.doi.org/10.1007/978-3-319-03780-6_14, doi:10.1007/978-3-319-03780-6_14.