CrossMark

On Group Feedback Vertex Set Parameterized by the Size of the Cutset

**Marek Cygan · Marcin Pilipczuk ·
Michał Pilipczuk**

Abstract We study parameterized complexity of a generalization of the classical FEEDBACK VERTEX SET problem, namely the GROUP FEEDBACK VERTEX SET problem: we are given a graph G with edges labeled with group elements, and the goal is to compute the smallest set of vertices that hits all cycles of G that evaluate to a non-null element of the group. This problem generalizes not only FEEDBACK VERTEX SET, but also SUBSET FEEDBACK VERTEX SET, MULTIWAY CUT and ODD CYCLE TRANSVERSAL. Completing the results of Guillemot (Discrete Optim 8(1):61–71, 2011), we provide a fixed-parameter algorithm for the parameterization by the size of the cutset only. Our algorithm works even if the group is given as a blackbox performing group operations.

M. Cygan · Michał Pilipczuk
Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
e-mail: cygan@mimuw.edu.pl

Michał Pilipczuk
e-mail: michal.pilipczuk@mimuw.edu.pl

Marcin Pilipczuk (✉)
Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK
e-mail: malcin@mimuw.edu.pl

1 Introduction

The parameterized complexity is an approach for tackling NP-hard problems by designing algorithms that perform well, when the instance is in some sense simple; its difficulty is measured by an integer, called the *parameter*, additionally appended to the input. Formally, we say that a problem is *fixed-parameter tractable* (FPT), if it admits an algorithm that given input of length n and parameter k, resolves the task in time $f(k)n^c$, where f is some computable function and c is a constant independent of the parameter.

The search for fixed-parameter algorithms led to the development of a number of new techniques and gave valuable insight into structures of many classes of NP-hard problems. Among them, there is a family of so-called *graph cut* problems, where the goal is to delete as few as possible edges or vertices (depending on the variant) in order to make the graph satisfy a global separation requirement. This class is perhaps best represented by the classical FEEDBACK VERTEX SET problem (FVS) where, given an undirected graph G, we seek for a minimum set of vertices that hits all cycles of G. Other examples are MULTIWAY CUT (MWC: separate each pair from a given set of terminals in a graph with a minimum cutset) or ODD CYCLE TRANSVERSAL (OCT: make a graph bipartite by a minimum number of vertex deletions).

The research on the aforementioned problems had a great impact on the development of parameterized complexity. The long line of research concerning parameterized algorithms for FVS contains [1–4,11–13,15,18,23], leading to an algorithm working in $O(3^k n^{O(1)})$ time [7]. The search for a polynomial kernel for FVS lead to surprising applications of deep combinatorial results such as Gallai's theorem [26], which has also been found useful in designing FPT algorithms [10]. While investigating the graph cut problems such as MWC, Márx [21] introduced the *important separator* technique, which turned out to be very robust and is now the key ingredient in parameterized algorithms for various problems such as variants of FVS [5,10] or ALMOST 2- SAT [24]. Moreover, the recent developments on MWC show applicability of linear programming in parameterized complexity, leading to the fastest currently known algorithms not only for MWC, but also ALMOST 2- SAT and OCT [9,22]. Last but not least, the research on the OCT problem resulted in the introduction of iterative compression, a simple yet powerful technique for designing parameterized algorithms [25].

1.1 Considered Problem

In this paper we study a generalization of the FVS problem, namely GROUP FEEDBACK VERTEX SET.[1] Let Σ be a finite (not necessarily abelian) group, with unit element 1_Σ. We use the multiplicative convention for denoting the group operation.

---

[1] In this paper, we follow standard graph notation in generality, and the notation of Guillemot [14] for group-labeled graphs. In particular, we use uv for undirected edge between vertices u and v, and (u, v) for a directed one pointed from u to v.

Definition 1 For a finite group Σ, a directed graph $G = (V, A)$ and a labeling function $\Lambda : A \to \Sigma$, we call (G, Λ) a Σ-*labeled* graph iff for each arc $(u, v) \in A$ we have $(v, u) \in A$ and $\Lambda((u, v)) = \Lambda((v, u))^{-1}$.

We somehow abuse the notation and by $(G \backslash X, \Lambda)$ denote the Σ-labeled graph $(G \backslash X, \Lambda|_{A(G \backslash X)})$, even though formally Λ has in its domain arcs that do not exist in $G \backslash X$ (i.e., we ignore that, to be completely formal, we need to restrict the domain of the labeling Λ).

For a path[2] $P = (v_1, \ldots, v_\ell)$ we denote $\Lambda(P) = \Lambda((v_1, v_2)) \cdot \ldots \cdot \Lambda((v_{\ell-1}, v_\ell))$. Similarly, for a cycle $C = (v_1, \ldots, v_\ell, v_1)$ we denote $\Lambda(C) = \Lambda((v_1, v_2)) \cdot \ldots \cdot \Lambda((v_{\ell-1}, v_\ell)) \cdot \Lambda((v_\ell, v_1))$. We call a cycle C a *non-null* cycle, iff $\Lambda(C) \neq 1_\Sigma$. Observe that if the group Σ is non-abelian, then it may happen that cyclic shifts of the same cycle yield different elements of the group; nevertheless, the notion of a non-null cycle is well-defined, as either all of them are equal to 1_Σ or none of them are.

Lemma 1 *Assume that* $(x_1, \ldots, x_\ell, x_1)$ *is a cycle in a* Σ-*labeled graph* (G, Λ). *If* $\Lambda((x_1, \ldots, x_\ell, x_1)) \neq 1_\Sigma$, *then* $\Lambda((x_2, \ldots, x_\ell, x_1, x_2)) \neq 1_\Sigma$.

Proof Let $g_1 = \Lambda((x_1, x_2))$ and $g_2 = \Lambda((x_2, \ldots, x_\ell, x_1))$. We have that $g_1 \cdot g_2 = 1_\Sigma$ iff $g_2 \cdot g_1 = 1_\Sigma$ and the lemma follows.   □

In the GROUP FEEDBACK VERTEX SET problem we want to hit all non-null cycles in a Σ-labeled graph using at most k vertices.

GROUP FEEDBACK VERTEX SET (GFVS)
Parameter: k.
Input: A Σ-labeled graph (G, Λ) and an integer k.
Question: Does there exist a set $X \subseteq V(G)$ of at most k vertices, such that there are no non-null cycles in $(G \backslash X, \Lambda)$?

As observed in [14], for a graph excluding a non-null cycle we can define a consistent labeling.

Definition 2 For a Σ-labeled graph (G, Λ) we call $\lambda : V \to \Sigma$ a *consistent labeling* iff for each arc $(u, v) = a \in A(G)$ we have $\lambda(v) = \lambda(u) \cdot \Lambda(a)$.

Lemma 2 ([14]) *A* Σ-*labeled graph* (G, Λ) *has a consistent labeling iff it does not contain a non-null cycle. Moreover, there is a polynomial-time algorithm which, given* (G, Λ), *finds either a non-null cycle in* G *or a consistent labeling of* G.

Note that when analysing the complexity of the GFVS problem, it is important how the group Σ is represented. In [14] it is assumed that Σ is given via its multiplication table as a part of the input. In this paper we assume a more general model, where operations in Σ are computed by a given blackbox. More precisely, we assume that we are given subroutines that can multiply two elements, return an inverse of an element, provide the neutral element 1_Σ, or check whether two elements are equal. The running times of our algorithms are always measured in terms of basic operations

---

[2] In this paper, all paths and cycles are simple.

and group operations, while space complexity is measured in the number of bits and group elements stored.

As noted in [20], GFVS subsumes not only the classical FVS problem, but also OCT (with $\Sigma = \mathbb{Z}_2$) and MWC (with Σ being an arbitrary group of size not smaller than the number of terminals). We note that if Σ is given in the blackbox, GROUP FEEDBACK VERTEX SET subsumes also EDGE SUBSET FEEDBACK VERTEX SET, which is equivalent to SUBSET FEEDBACK VERTEX SET [10].

EDGE SUBSET FEEDBACK VERTEX SET (ESFVS)
Parameter: k.
Input: An undirected graph G, a set $S \subseteq E(G)$ and an integer k.
Question: Does there exist a set $X \subseteq V(G)$ of at most k vertices, such that in $G \backslash X$ there are no cycles with at least one edge from S?

Lemma 3 *Given an* ESFVS *instance* (G, S, k)*, one can in polynomial time construct an equivalent* GFVS *instance* (G', Λ, k) *with group* $\Sigma = \mathbb{Z}_2^{|S|}$.

Proof To construct the new GFVS instance, create the graph G' by replacing each edge of G with arcs in both direction, keep the parameter k, take $\Sigma = \mathbb{Z}_2^{|S|}$ and construct a Σ-labeling Λ by setting any $|S|$ linearly independent values of $\Lambda((u, v))$ for $uv \in S$ and $\Lambda((u, v)) = 1_\Sigma$ for $uv \notin S$. Clearly, this construction can be done in polynomial time and the operations on the group Σ can be performed by a subroutine in time polynomial in $|S|$ by representing elements of Σ as bit vectors of length $|S|$. $\qquad\square$

We note that the GROUP FEEDBACK VERTEX SET problem was also studied from the graph theoretical point of view, as, in addition to the aforementioned reductions, it also subsumes the setting of Mader's $\mathcal{S}$-paths theorem [6,19]. In particular, Kawarabayashi and Wollan proved the Erdős–Pósa property for non-null cycles in highly connected graphs, generalizing a list of previous results [19], while Wollan proved the Erdős–Pósa property for non-null cycles in general graphs labeled with an abelian group [28].

The study of parameterized complexity of GFVS was initiated by Guillemot [14], who presented a fixed-parameter algorithm for GFVS parameterized by $|\Sigma| + k$ running in time[3] $O^*(2^{O(k \log |\Sigma|)})$. When parameterized by k, Guillemot showed a fixed-parameter algorithm for the easier edge-deletion variant of GFVS, running in time $O^*(2^{O(k \log k)})$. Recently, Kratsch and Wahlström presented a randomized kernelization algorithm that reduces the size of a GFVS instance to $O(k^{2|\Sigma|})$ [20].

Before we proceed to the description of our results, let us briefly sketch their motivation. The main purpose of studying the GFVS problem is to find the common points in the fixed-parameter algorithms for problems it generalizes. Precisely this approach has been presented by Guillemot in [14], where at the base of the algorithm lies a subroutine that solves a very general version of MULTIWAY CUT. When reducing various graph cut problems to GFVS, usually the size of the group depends on the number of distinguished vertices or edges in the instance, as in

---

[3] The $O^*()$ notation suppresses terms polynomial in the input size.

Lemma 3. Hence, an application of the general $O^*(2^{O(k \log |\Sigma|)})$ algorithm of Guillemot unfortunately incorporates this parameter in the running time. It appears that by a more refined combinatorial analysis, usually one can get rid of this dependence; this is the case both in SUBSET FEEDBACK VERTEX SET [10] and in MULTIWAY CUT [9,22]. This suggested that the phenomenon can be, in fact, more general.

1.2 Our Result and Techniques

Our main result is a fixed-parameter algorithm for GFVS parameterized by the size of the cutset only. Recall that time and space complexities refer to basic and group operations performed, and bits and group elements stored, respectively.

Theorem 1 GROUP FEEDBACK VERTEX SET *can be solved in* $O^*(2^{O(k \log k)})$ *time and polynomial space.*

Our algorithm uses a similar approach as described by Kratsch and Wahlström in [20]: in each step of iterative compression, when we are given a solution Z of size $k + 1$, we guess the values of a consistent labeling on the vertices of Z, and reduce the problem to MULTIWAY CUT. However, by a straightforward application of this approach we obtain $O^*(2^{O(k \log |\Sigma|)})$ time complexity. To reduce the dependency on $|\Sigma|$, we carefully analyse the structure of a solution, provide a few reduction rules in a spirit of the ones used in the recent algorithm for SUBSET FEEDBACK VERTEX SET [10] and, finally, for each vertex of Z we reduce the number of choices for a value of a consistent labeling to polynomial in k. Therefore, the number of reasonable consistent labelings of Z is bounded by $2^{O(k \log k)}$ and we can afford solving a MULTIWAY CUT instance for each such labeling.

A comparison with the previous algorithm for SUBSET FEEDBACK VERTEX SET [10] is in place; note that the bound on the running time of our algorithm matches the one for SUBSET FEEDBACK VERTEX SET [10], while our algorithm works in a much more general framework. In our opinion, the group-labeled setting is a much more convenient and insightful way of looking at graph-separation problems as compared to the definition of SUBSET FEEDBACK VERTEX SET. To support this claim, let us briefly analyse how the algorithm of Theorem 1 solves an ESFVS instance (G, S, k), via the reduction of Lemma 3. Every edge of S translates to a different $\mathbb{Z}_2$ coordinate of the group $\Sigma = \mathbb{Z}_2^{|S|}$. The step where we guess the correct labeling of every vertex of the modulator Z corresponds, in the language of ESFVS, to a guessing, between every pair u, v of vertices of Z, which edges of S lie on surviving paths between u and v — and the core of our argumentation is that there is only a limited number of reasonable choices. Such a branching step, while completely natural in the GFVS language, seems far-fetched and hard to guess in the ESFVS regime, in particular if one needs to argue about a limited number of subcases.

We would also like to mention that running time bound of both the algorithm of [10] and the algorithm of Theorem 1 have high polynomial dependency on the input size, due to the usage of iterative compression and multiple flow computations, and in this work we do not analyse this dependency in detail.

Finally, let us mention that after this work has been presented at WG 2012 [8], Wahlström [27] has shown how to generalize the LP-branching approach introduced in [9] to a number of graph separation problems and, among other results, he presented an $O^*(4^k)$-time algorithm for GFVS. At the same time, Iwata et al. [16] developed a way to perform several similar flow-based branching algorithms keeping only linear dependence on the input size in the running time bound. Their techniques turned out to be compatible with Wahlström's approach [17], yielding a linear dependency on the input size in time bounds for all *edge deletion* problems considered in [27].

2 Preliminaries

2.1 Notation

We use standard graph notation. For a graph G, by $V(G)$ and $E(G)$ we denote its vertex and edge sets, respectively, and uv denotes an edge between vertices u and v. In case of a directed graph G, we denote the arc set of G by $A(G)$, and (u, v) denotes an arc pointed from u to v. For $v \in V(G)$, its neighborhood $N_G(v)$ is defined as $N_G(v) = \{u : uv \in E(G)\}$, and $N_G[v] = N_G(v) \cup \{v\}$ is the closed neighborhood of v. We extend this notation to subsets of vertices: $N_G[X] = \bigcup_{v \in X} N_G[v]$ and $N_G(X) = N_G[X] \backslash X$. For a set $X \subseteq V(G)$ by $G[X]$ we denote the subgraph of G induced by X. For a set X of vertices or edges of G, by $G \backslash X$ we denote the graph with the vertices or edges of X removed; in case of vertex removal, we remove also all the incident edges. By somehow abusing the notation, we often treat the (directed) Σ-labeled graph also as an undirected graph, as the neighborhood relation in the underlying undirected graph is the same.

In the GROUP FEEDBACK VERTEX SET problem definition in [14] a set of forbidden vertices $F \subseteq V(G)$ is additionally given as a part of the input. One can easily gadget such vertices by replacing each of them by a clique of size $k + 1$ labeled with 1_Σ; therefore, for the sake of simplicity we assume that all the vertices are allowed.

3 Algorithm

In this section we prove Theorem 1. We proceed with a standard application of the iterative compression technique in Sect. 3.1. In each step of the iterative compression, we solve a COMPRESSION GROUP FEEDBACK VERTEX SET problem, where we are given a solution Z of size a bit too large — $k + 1$ — and we are to find a new solution disjoint from it. We first prepare the COMPRESSION GROUP FEEDBACK VERTEX SET instance by *untangling* it in Sect. 3.2, in the same manner as it is done in the kernelization algorithm of [20]. The main step of the algorithm is done in Sect. 3.3, where we provide a set of reduction rules that enable us for each vertex $v \in Z$ to limit the number of choices for a value of a consistent labeling on v to polynomial in k. Finally, we iterate over all $O^*(2^{O(k \log k)})$ remaining labelings of Z and, for each labeling, reduce the instance to MULTIWAY CUT (Sect. 3.4).

3.1 Iterative Compression

The first step in the proof of Theorem 1 is a standard technique in the design of parameterized algorithms, that is, iterative compression, introduced by Reed et al. [25]. Iterative compression was also the first step of the parameterized algorithm for SUBSET FEEDBACK VERTEX SET [10].

We define a *compression problem*, where the input additionally contains a feasible solution $Z \subseteq V(G)$, and we are asked whether there exists a solution of size at most k which is disjoint from Z.

COMPRESSION GROUP FEEDBACK VERTEX SET (C-GFVS)
Parameter: $k + |Z|$.
Input: A Σ-labeled graph (G, Λ), an integer k and a set $Z \subseteq V(G)$, such that $(G \backslash Z, \Lambda)$ has no non-null cycle.
Goal: Find a set $X \subseteq V(G) \backslash Z$ of at most k vertices, such that there is no non-null cycle in $(G \backslash X, \Lambda)$ or return NO, if such a set does not exist.

In Sect. 3.2 we prove the following lemma providing a parameterized algorithm for COMPRESSION GROUP FEEDBACK VERTEX SET.

Lemma 4 *The* COMPRESSION GROUP FEEDBACK VERTEX SET *problem can be solved in* $O^*(2^{O(|Z|(\log k + \log |Z|))} \cdot 2^k)$ *time and polynomial space.*

Armed with the aforementioned result, we can easily prove Theorem 1.

Proof (*of Theorem* 1) In the iterative compression approach we start with an empty solution for an empty graph, and in each of the n steps we add a single vertex both to a feasible solution and to the graph; we use Lemma 4 to compress the feasible solution after guessing which vertices of the solution of size at most $k + 1$ should not be removed.

Formally, for a given instance $(G = (V, A), \Lambda, k)$ let $V = \{v_1, \ldots, v_n\}$. For $0 \leq i \leq n$ define $V_i = \{v_1, \ldots, v_i\}$ (in particular, $V_0 = \emptyset$) and let Λ_i be the function Λ restricted to the set of arcs $A_i = \{(u, v) \in A : u, v \in V_i\}$. Initially we set $X_0 = \emptyset$, which is a solution to the graph $(G[V_0], \Lambda_0)$. For each $i = 1, \ldots, n$ we set $Z_i = X_{i-1} \cup \{v_i\}$, which is a feasible solution to $(G[V_i], \Lambda_i)$ of size at most $k + 1$. If $|Z_i| \leq k$, then we set $X_i = Z_i$ and continue the inductive process. Otherwise, if $|Z_i| = k + 1$, we apply Lemma 4 to the instance $I_{Z_i'} = (G[V_i \backslash (Z_i \backslash Z_i')], \Lambda_i, k' = |Z_i'| - 1, Z_i')$ for every choice of $Z_i' \subseteq Z_i$. If for each set Z_i' the algorithm from Lemma 4 returns NO, then there is no solution for $(G[V_i], \Lambda_i)$ and, consequently, there is no solution for (G, Λ). However, if for some Z_i' the algorithm from Lemma 4 returns a set X_i' of size smaller than $|Z_i'|$, then we set $X_i = (Z_i \backslash Z_i') \cup X_i'$. Since $|X_i| = |Z_i \backslash Z_i'| + |X_i'| < |Z_i| = k + 1$, the set X_i is a solution of size at most k for the instance (G_i, Λ_i).

Finally, we observe that since $(G_n, \Lambda_n) = (G, \Lambda)$, the set X_n is a solution for the initial instance $(G = (V, A), \Lambda, k)$ of GROUP FEEDBACK VERTEX SET. The claimed bound on the running time follows from the observation that $|Z_i| \leq k + 1$ for each of polynomially many steps. $\square$

At this point a reader might wonder why we do not add an assumption $|Z| \leq k+1$ to the C- GFVS problem definition and parameterize the problem solely by k. The reason for this is that in Sect. 3.3 we will solve the C- GFVS problem recursively, sometimes decreasing the value of k without decreasing the size of Z, and to always work with a feasible instance of the C- GFVS problem we avoid adding the $|Z| \leq k+1$ assumption to the problem definition.

3.2 Untangling

In order to prove Lemma 4 we use the concept of *untangling*, previously used by Kratsch and Wahlström [20]. We transform an instance of C- GFVS to ensure that each arc (u, v) with both endpoints in $V(G) \backslash Z$ is labeled 1_Σ by Λ.

Definition 3 We call an instance $(G = (V, A), \Lambda, k, Z)$ of C- GFVS *untangled*, iff for each arc $(u, v) \in A$ such that $u, v \in V \backslash Z$ we have $\Lambda((u, v)) = 1_\Sigma$.

Moreover, by *untangling* a labeling Λ around vertex x with a group element g we mean changing the labeling to $\Lambda' : A \to \Sigma$, such that for $(u, v) = a \in A$, we have

$$\Lambda'(a) = \begin{cases} g \cdot \Lambda(a) & \text{if } u = x; \\ \Lambda(a) \cdot g^{-1} & \text{if } v = x; \\ \Lambda(a) & \text{otherwise.} \end{cases}$$

Lemma 5 *Let $(G = (V, A), \Lambda)$ be a Σ-labeled graph, $x \in V$ be a vertex of G and let $g \in \Sigma$ be a group element. For every subset of vertices $X \subseteq V$ the graph $(G \backslash X, \Lambda)$ contains a non-null cycle iff $(G \backslash X, \Lambda')$ contains a non-null cycle, where Λ' is the labeling Λ untangled around the vertex x with a group element g.*

Proof The lemma follows from the fact that for any cycle C in G we have $\Lambda(C) = 1_\Sigma$ iff $\Lambda'(C) = 1_\Sigma$. $\qquad\square$

In Sect. 3.3 we prove the following lemma.

Lemma 6 *The* COMPRESSION GROUP FEEDBACK VERTEX SET *problem for untangled instances can be solved in $O^*(2^{O(|Z|(\log k + \log |Z|))} \cdot 2^k)$ time and polynomial space.*

Having Lemmata 5 and 6 we can prove Lemma 4.

Proof (*of Lemma 4*) Let (G, Λ, k, Z) be an instance of C- GFVS. Since $(G \backslash Z)$ has no non-null cycle, by Lemma 2 there is a consistent labeling λ of $(G \backslash Z, \Lambda)$.

Let Λ' be a result of untangling Λ around each vertex $v \in V(G) \backslash Z$ with $\lambda(v)$. Clearly, this operation can be done in polynomial time. Note that, by associativity of Σ, the order in which we untangle subsequent vertices does not matter. After all the untangling operations, for an arc $a = (u, v) \in A(G)$, such that $u, v \in V(G) \backslash Z$, we have $\Lambda'(a) = (\lambda(u) \cdot \Lambda(a)) \cdot \lambda(v)^{-1} = \lambda(v) \cdot \lambda(v)^{-1} = 1_\Sigma$. Therefore, by Lemma 5 the instance (G, Λ', k, Z) is an untangled instance of C- GFVS, which is a YES-instance iff (G, Λ, k, Z) is a YES-instance. Consequently, we can use Lemma 6 and the claim follows. $\qquad\square$

3.3 Fixing a Labeling on Z

In this section we prove Lemma 6 using the following lemma, which we prove in Sect. 3.4.

Lemma 7 *Let (G, Λ, k, Z) be an untangled instances of* C- GFVS. *There is an algorithm which for a given function $\phi : Z \to \Sigma$, finds a set $X \subseteq V(G) \backslash Z$ of size at most k, such that there exists a consistent labeling $\lambda : V(G) \backslash X \to \Sigma$ of $(G \backslash X, \Lambda)$, where $\lambda|_Z = \phi$, or checks that such a set X does not exist; the algorithm works in $O^*(2^k)$ time and uses polynomial space.*

We could try all $(|\Sigma| + 1)^{|Z|}$ possible assignments ϕ and use the algorithm from Lemma 7. Unfortunately, since $|\Sigma|$ is not our parameter we cannot iterate over all such assignments. Therefore, the goal of this section is to show that after some preprocessing, it is enough to consider only $2^{O(|Z|(\log k + \log |Z|))}$ assignments ϕ; together with Lemma 2 and Lemma 7 this suffices to prove Lemma 6.

Definition 4 Let (G, Λ, k, Z) be an untangled instance of C- GFVS, let z be a vertex in Z and by Σ_z denote the set $\Lambda(\{(z, v) \in A(G) : v \in V(G) \backslash Z\})$.

By a *flow graph* $F(G, \Lambda, Z, z)$, we denote the undirected graph (V', E'), where $V' = (V(G) \backslash Z) \cup \Sigma_z$ and $E' = \{uv : (u, v) \in A(G[V(G) \backslash Z])\} \cup \{gv : (z, v) \in A(G), v \in V(G) \backslash Z, \Lambda((z, v)) = g\}$.

Less formally, in the flow graph we take the underlying undirected graph of $G[V(G) \backslash Z]$ and add a vertex for each group element $g \in \Sigma_z$, that is a group element for which there exists an arc from z to $V(G) \backslash Z$ labeled with g by Λ. A vertex $g \in \Sigma_z$ is adjacent to all the vertices of $V(G) \backslash Z$ for which there exists an arc going from z, labeled with g by Λ.

Lemma 8 *Let (G, Λ, k, Z) be an untangled instance of* C- GFVS. *Let H be the flow graph $F(G, \Lambda, Z, z)$ for some $z \in Z$. If for some vertex $v \in V(H) \backslash \Sigma_z$, there are at least $k + 2$ paths in H from v to Σ_z that are vertex disjoint apart from v, then v belongs to every solution of* C- GFVS.

Proof Let us assume that v is not a part of a solution $X \subseteq V(G) \backslash Z$, where $|X| \leq k$. Then at least 2 out of the $k + 2$ paths from v to Σ_z remain in $H \backslash X$. These two paths are vertex disjoint apart from v and end in different elements of Σ_z, so they correspond to a non-null cycle in $G \backslash X$, a contradiction. $\square$

Definition 5 For an untangled instance (G, Λ, k, Z) of C- GFVS by an *external path* we denote any path P beginning and ending in Z, but with all internal vertices belonging to $V(G) \backslash Z$. Moreover, for two distinct vertices $z_1, z_2 \in Z$, by $\Sigma(z_1, z_2)$ we denote the set of all elements $g \in \Sigma$, for which there exists an external path P from z_1 to z_2 with $\Lambda(P) = g$.

Note that an arc (z_1, z_2) for $z_1, z_2 \in Z$ also forms an external path from z_1 to z_2.

Lemma 9 *Let (G, Λ, k, Z) be an untangled instance of* C- GFVS. *If for each $z \in Z$ and $v \in V(G) \backslash Z$ there are at most $k + 1$ vertex disjoint paths from v to Σ_z in $F(G, \Lambda, Z, z)$ and for some $z_1, z_2 \in Z, z_1 \neq z_2$, we have $|\Sigma(z_1, z_2)| \geq k^3(k+1)^2 + 2$, then there is no solution for (G, Λ, k, Z).*

Proof Let us assume that $X \subseteq V(G) \backslash Z$ is a solution for (G, Λ, k, Z); in particular, $|X| \leq k$. Let $\mathcal{P}$ be a set of external paths from z_1 to z_2, containing exactly one path P for each $g \in \Sigma(z_1, z_2)$ with $\Lambda(P) = g$. Note that the only arcs with non-null labels in P are possibly the first and the last arc.

By the pigeon-hole principle, there exists a vertex $w \in X$, which belongs to at least $k^2(k+1)^2 + 1$ paths in $\mathcal{P}$, since otherwise there would be at least two paths in $\mathcal{P}$ disjoint from X, creating a non-null closed walk disjoint from X. Note that existence of a non-null closed walk disjoint from X is a sufficient proof that X is not a solution to (G, Λ, k, Z), as it contradicts existence of a consistent labeling, guaranteed by Lemma 2.

Consider the connected component C of $G[V(G) \backslash Z]$ to which w belongs. Observe that there exists a vertex $z \in \{z_1, z_2\}$ that has at least $k(k+1) + 1$ incident arcs going to C with pairwise different labels in Λ, since otherwise v would belong to at most $k^2(k+1)^2$ paths in $\mathcal{P}$.

Let H be the flow graph $F(G, \Lambda, Z, z)$ and let $T \subseteq \Sigma_z$ be the set of labels of arcs going from z to C; recall that $|T| > k(k+1)$. Since there is no non-null cycle in $(G \backslash X, \Lambda)$, we infer that in $H_0 = H[C \cup T] \backslash (X \cap C)$, no two vertices of T belong to the same connected component. Moreover, as C is connected in G, for each $t \in T$ there exists a path P_t with endpoints w and t in $H[C \cup T]$. Let v_t be the closest to t vertex from X on the path P_t; note that such a vertex always exists, as $w \in X$. As $|X| \leq k$ and $|T| > k(k+1)$, there exists $v \in X$ such that $v = v_t$ for at least $k+2$ elements $t \in T$. By the definition of the vertices v_t and the fact that there are no two vertices of T in the same connected component of H_0, the subpaths of P_t from t to v_t for all t with $v = v_t$ are vertex disjoint apart from v. As there are at least $k+2$ of them, we have a contradiction.   □

We are now ready to prove Lemma 6 given Lemma 7.

Proof (*of Lemma* 6) If there exists a vertex v satisfying the properties of Lemma 8, we can assume that it has to be a part of the solution; therefore, we can remove the vertex from the graph and solve the problem for decremented parameter value. Hence, we assume that for each $z \in Z$ and $v \in V(G) \backslash Z$, there are at most $k + 1$ vertex disjoint paths from v to Σ_z in $F(G, \Lambda, Z, z)$. We note that one can compute the number of such vertex disjoint paths in polynomial time, using a maximum flow algorithm.

Observe that one can easily compute the sets $\Sigma(z_1, z_2)$ for every $z_1, z_2 \in Z$, since the only non-null label arcs on paths contributing to $\Sigma(z_1, z_2)$ are the first and the last one, and we can iterate over all such arcs and check whether their endpoints are in the same connected component in $G[V(G) \backslash Z]$. Clearly, this can be done in polynomial time.

By Lemma 9, if there is a pair of vertices $z_1, z_2 \in Z$ with $|\Sigma(z_1, z_2)| \geq k^3(k+1)^2 + 2$, we know that there is no solution. Otherwise, we can enumerate all the reasonable labelings of Z as follows. For the sake of analysis let $G' = (Z, E')$ be an auxiliary undirected graph, where two vertices of Z are adjacent, when they are connected by an external path in $G \backslash X$, for some fixed solution $X \subseteq V(G) \backslash Z$. Let F be any spanning forest of G'. Since F has at most $|Z| - 1$ edges, we can guess F, by trying at most

$$\sum_{i=0}^{|Z|-1} \binom{\binom{|Z|}{2}}{i} = 2^{O(|Z| \log |Z|)}$$

possibilities; in the above expression we iterate over all possible choices of $i = |E(F)|$, and then choose i edges out of at most $\binom{|Z|}{2}$ options.

Let us assume, that we have guessed F correctly. Observe that for any two vertices $z_1, z_2 \in Z$, belonging to two different connected components of F, there is no path between z_1 and z_2 in $G \setminus X$. Therefore, there exists a consistent labeling of $G \setminus X$, which labels an arbitrary vertex from each connected component of F with 1_Σ. Having fixed the labeling on one vertex from each component of F, we can root the components in corresponding vertices and iteratively guess the labeling on the remaining vertices in a top-down manner. At each step we use the fact that if we have already fixed a value $\phi(z_1)$, then for each external path corresponding to an edge $z_1 z_2$ of F, there are at most $k^3(k+1)^2 + 1$ possible values of $\phi(z_2)$, since $\phi^{-1}(z_1) \cdot \phi(z_2) \in \Sigma(z_1, z_2)$; recall that we have already computed the sets $\Sigma(z_1, z_2)$ earlier in the process. Hence, having fixed F there are at most $2^{O(|Z| \log k)}$ possible labelings ϕ of Z, as for each edge of the forest F we choose one of at most $k^3(k+1)^2 + 1$ options. As the number of choices of F is bounded by $2^{O(|Z| \log |Z|)}$, we obtain at most $2^{O(|Z|(\log k + \log |Z|))}$ labelings ϕ of Z in total, and we can use Lemma 7 for each of them. □

3.4 Reduction to Multiway Cut

In this section, we prove Lemma 7, by a reduction to MULTIWAY CUT. A similar reduction was also used recently by Kratsch and Wahlström in the kernelization algorithm for GROUP FEEDBACK VERTEX SET parameterized by k with constant $|\Sigma|$ [20]. Currently the fastest FPT algorithm for MULTIWAY CUT is due to Cygan et al. [9], and it solves the problem in $O^*(2^k)$ time and polynomial space.

MULTIWAY CUT
Parameter: k.
Input: An undirected graph $G = (V, E)$, a set of terminals $T \subseteq V$, and a positive integer k.
Goal: Find a set $X \subseteq V \setminus T$, such that $|X| \leq k$ and no pair of terminals from the set T is contained in one connected component of the graph $G[V \setminus X]$, or return NO if such a set X does not exist.

Proof (of Lemma 7) Firstly, we check whether the given function ϕ satisfies $\phi(z_2) = \phi(z_1) \cdot \Lambda((z_1, z_2))$, for each arc $(z_1, z_2) \in G[Z]$, since otherwise there is no set X we are looking for.

Given a Σ-labeled graph (G, Λ), a set Z, an integer k, and a function $\phi : Z \to \Sigma$, we create an undirected graph $G' = (V', E')$. As the vertex set, we set $V' = (V(G) \setminus Z) \cup T$ and $T = \{g : (u, v) \in A(G), u \in Z, v \in V(G) \setminus Z, \phi(u) \cdot \Lambda((u, v)) = g\}$. Note that the set $T \cup \{1_\Sigma\}$ is exactly the set of potential labels for vertices in $V(G) \setminus Z$ of a consistent labeling of (G, Λ) which extends ϕ on Z. Observe that we do not need to consider here arcs from $V(G) \setminus Z$ to Z, as in the definition of a Σ-labeled graph we have assumed that every arc (u, v) with label $\Lambda((u, v))$ is

accompanied with an arc (v, u) with label $\Lambda((v, u)) = \Lambda((u, v))^{-1}$. As the edge set, we set $E' = \{uv : (u, v) \in A(G[V(G)\backslash Z])\} \cup \{gv : (u, v) \in A(G),\ u \in Z,\ v \in V(G)\backslash Z,\ \phi(u) \cdot \Lambda((u, v)) = g\}$. We show that (G', T, k) is a YES-instance of MULTIWAY CUT iff there exists a set $X \subseteq V(G)\backslash Z$, such that there exists a consistent labeling λ of $(G\backslash X, \Lambda)$ with $\lambda|_Z = \phi$.

Let X be a solution for (G', T, k). We define a consistent labeling λ of $(G\backslash X, \Lambda)$. For $v \in Z$ we set $\lambda(v) = \phi(v)$. For $v \in (V(G)\backslash Z)\backslash X$, if v is reachable from a terminal $g \in T$ in $G'\backslash X$, we set $\lambda(v) = g$. If $v \in (V(G)\backslash Z)\backslash X$ is not reachable from any terminal in G', we set $\lambda(v) = 1_\Sigma$. Since each arc in $A(G[V(G)\backslash Z])$ is labeled 1_Σ by Λ, and each vertex in $V(G)\backslash Z$ is reachable from at most one terminal in $G'\backslash X$, λ is a consistent labeling of $(G\backslash X, \Lambda)$.

Let $X \subseteq V(G)\backslash Z$ be a set of vertices of G, $|X| \le k$, such that there is a consistent labeling λ of $(G\backslash X, \Lambda)$, where $\lambda|_Z = \phi$. By the definition of edges between T and $V(G)\backslash Z$ in G', each vertex of $V(G)\backslash Z$ is reachable from at most one terminal in G', since otherwise λ would not be a consistent labeling of $(G\backslash X, \lambda)$. Therefore, X is a solution for (G', T, k).

We can now apply the algorithm for MULTIWAY CUT of [9] to the instance (G', T, k) in order to conclude the proof. □

4 Conclusions

We have shown a relatively simple fixed-parameter algorithm for GROUP FEEDBACK VERTEX SET running in time $O^*(2^{O(k \log k)})$. Our algorithm works even in a robust blackbox model, that allows us to generalize the recent algorithm for SUBSET FEEDBACK VERTEX SET [10] within the same complexity bound. In a subsequent work, Wahlström [27] improved the dependency on the parameter in the running time to 4^k, using a significantly different approach.

We would like to note that if we represent group elements by strings consisting of g and g^{-1} for $g \in \Lambda(A(G))$ (formally, we perform the computations in the free group over generators corresponding to the arcs of the graph), then after slight modifications of our algorithm we can solve the GROUP FEEDBACK VERTEX SET problem even for infinite groups for which the word problem, i.e., the problem of checking whether results of two sequences of multiplications are equal, is polynomial-time solvable. The lengths of representations of group elements created during the computation can be bounded linearly in the size of the input graph. Therefore, if a group admits a polynomial-time algorithm solving the word problem, then we can use this algorithm as the blackbox.

References

1. Becker, A., Bar-Yehuda, R., Geiger, D.: Randomized algorithms for the loop cutset problem. J. Artif. Intell. Res. (JAIR) **12**, 219–234 (2000)

2. Bodlaender, H.L.: On disjoint cycles. Int. J. Found. Comput. Sci. **5**(1), 59–68 (1994)
3. Cao, Y., Chen, J., Liu, Y.: On feedback vertex set new measure and new structures. In: Kaplan, H. (ed.) SWAT, Lecture Notes in Computer Science, vol. 6139, pp. 93–104. Springer, Berlin (2010)
4. Chen, J., Fomin, F.V., Liu, Y., Lu, S., Villanger, Y.: Improved algorithms for feedback vertex set problems. J. Comput. Syst. Sci. **74**(7), 1188–1198 (2008)
5. Chen, J., Liu, Y., Lu, S., O'Sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem. J. ACM **55**(5), 1–19 (2008)
6. Chudnovsky, M., Geelen, J., Gerards, B., Goddyn, L.A., Lohman, M., Seymour, P.D.: Packing non-zero A-paths in group-labelled graphs. Combinatorica **26**(5), 521–532 (2006)
7. Cygan M, Nederlof J, Pilipczuk M, Pilipczuk M, van Rooij JMM, Wojtaszczyk JO (2011) Solving connectivity problems parameterized by treewidth in single exponential time. In Ostrovsky R (ed) FOCS, pp. 150–159. IEEE (2011)
8. Cygan, M., Pilipczuk, M., Pilipczuk, M.: On group feedback vertex set parameterized by the size of the cutset. In: Golumbic, M.C., Stern, M., Levy, A., Morgenstern, G. (eds.) WG, Lecture Notes in Computer Science, vol. 7551, pp. 194–205. Springer, Berlin (2012)
9. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: On multiway cut parameterized above lower bounds. ACM Trans. Comput. Theory **5**(1), 3 (2013)
10. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: Subset feedback vertex set is fixed-parameter tractable. SIAM J. Discrete Math. **27**(1), 290–309 (2013)
11. Dehne, F.K.H.A., Fellows, M.R., Langston, M.A., Rosamond, F.A., Stevens, K.: An $O(2^{o(k)})n^3$ FPT algorithm for the undirected feedback vertex set problem. Theory Comput. Syst. **41**(3), 479–492 (2007)
12. Downey, R.G., Fellows, M.R.: Fixed parameter tractability and completeness. In: Complexity Theory: Current Research, Dagstuhl Workshop, pp. 191–225 (1992)
13. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Heidelberg (1999)
14. Guillemot, S.: FPT algorithms for path-transversal and cycle-transversal problems. Discret. Optim. **8**(1), 61–71 (2011)
15. Guo, J., Gramm, J., Hüffner, F., Niedermeier, R., Wernicke, S.: Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. J. Comput. Syst. Sci. **72**(8), 1386–1396 (2006)
16. Iwata, Y., Oka, K., Yoshida, Y.: Linear-time FPT algorithms via network flow. In: Chekuri, C. (ed.) SODA, pp. 1749–1761. SIAM (2014)
17. Iwata, Y., Wahlström, M., Yoshida, Y.: Half-integrality, LP-branching and FPT algorithms. CoRR abs/1310.2841v2 (2014)
18. Kanj, I.A., Pelsmajer, M.J., Schaefer, M.: Parameterized algorithms for feedback vertex set. In: Downey, R.G., Fellows, M.R., Dehne, F.K.H.A. (eds.) IWPEC, Lecture Notes in Computer Science, vol. 3162, pp. 235–247. Springer, Berlin (2004)
19. Kawarabayashi, K., Wollan, P.: Non-zero disjoint cycles in highly connected group labelled graphs. J. Comb. Theory Ser. B **96**(2), 296–301 (2006)
20. Kratsch, S., Wahlström, M.: Representative sets and irrelevant vertices: New tools for kernelization. In: FOCS, pp. 450–459. IEEE Computer Society (2012)
21. Marx, D.: Parameterized graph separation problems. Theor. Comput. Sci. **351**(3), 394–406 (2006)
22. Narayanaswamy, N.S., Raman, V., Ramanujan, M.S., Saurabh, S.: LP can be a cure for parameterized problems. In: Dürr, C., Wilke, T. (eds.) STACS, LIPIcs, vol. 14, pp. 338–349. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2012)
23. Raman, V., Saurabh, S., Subramanian, C.R.: Faster fixed parameter tractable algorithms for finding feedback vertex sets. ACM Trans. Algorithms **2**(3), 403–415 (2006)
24. Razgon, I., O'Sullivan, B.: Almost 2-SAT is fixed-parameter tractable. J. Comput. Syst. Sci. **75**(8), 435–450 (2009)
25. Reed, B.A., Smith, K., Vetta, A.: Finding odd cycle transversals. Oper. Res. Lett. **32**(4), 299–301 (2004)
26. Thomassé, S.: A $4k^2$ kernel for feedback vertex set. ACM Trans. Algorithms **6**(2), 32:1–32:8 (2010)
27. Wahlström, M.: Half-integrality, LP-branching and FPT algorithms. In: Chekuri, C. (ed.) SODA, pp. 1762–1781. SIAM (2014)
28. Wollan, P.: Packing cycles with modularity constraints. Combinatorica **31**(1), 95–126 (2011)