

# On group feedback vertex set parameterized by the size of the cutset

Marek Cygan \*

Marcin Pilipczuk<sup>†</sup>Michał Pilipczuk<sup>‡</sup>

## Abstract

We study the parameterized complexity of a robust 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 [Discr. Opt. 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 polynomial-time oracle.

## 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 a 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  $G$ . Another 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, 2, 3, 4, 10, 11, 12, 14, 16, 20], leading to an algorithm working in  $3^k n^{O(1)}$  time [7]. The search for a polynomial kernel for FVS lead to surprising applications of deep combinatorial results such as the Gallai's theorem [23], which has also been found useful in designing FPT algorithms [9]. While investigating the graph cut problems such as MWC, Márx [18] 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, 9] or ALMOST 2-SAT [21]. Moreover, the recent developments on MWC show applicability

---

\*Institute of Informatics, University of Warsaw, Poland, cygan@mimuw.edu.pl

†Institute of Informatics, University of Warsaw, Poland, malcin@mimuw.edu.pl

‡Department of Informatics, University of Bergen, Norway, michal.pilipczuk@ii.uib.no

of linear programming in parameterized complexity, leading to the fastest currently known algorithms not only for MWC, but also ALMOST 2-SAT and OCT [8, 19]. 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 [22].

**Considered problem.** In this paper we study a robust generalization of the FVS problem, namely GROUP FEEDBACK VERTEX SET<sup>1</sup>. Let  $\Sigma$  be a finite (not necessarily abelian) group, with unit element  $1_\Sigma$ . We use the multiplicative convention for denoting the group operation.

**Definition 1.** For a finite group  $\Sigma$ , a directed graph  $G = (V, A)$  and a labeling function  $\Lambda : A \rightarrow \Sigma$ , we call  $(G, \Lambda)$  a  $\Sigma$ -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  $(G \setminus X, \Lambda)$  denote the  $\Sigma$ -labeled graph  $(G, \Lambda)$  with vertices of  $X$  removed, even though formally  $\Lambda$  has in its domain arcs that do not exist in  $G \setminus X$ .

For a path  $P = (v_1, \dots, v_\ell)$  we denote  $\Lambda(P) = \Lambda((v_1, v_2)) \cdot \dots \cdot \Lambda((v_{\ell-1}, v_\ell))$ . Similarly, for a cycle  $C = (v_1, \dots, v_\ell, v_1)$  we denote  $\Lambda(C) = \Lambda((v_1, v_2)) \cdot \dots \cdot \Lambda((v_{\ell-1}, v_\ell)) \cdot \Lambda((v_\ell, v_1))$ . We call a cycle  $C$  a *non-null cycle*, iff  $\Lambda(C) \neq 1_\Sigma$ . Observe that if the group  $\Sigma$  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  $1_\Sigma$  or none of them.

**Lemma 2.** Let  $(x_1, \dots, x_\ell, x_1)$  be a cycle in a  $\Sigma$ -labeled graph  $(G, \Lambda)$ . If  $\Lambda((x_1, \dots, x_\ell, x_1)) \neq 1_\Sigma$ , then  $\Lambda((x_2, \dots, x_\ell, x_1, x_2)) \neq 1_\Sigma$ .

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

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

<p><b>GROUP FEEDBACK VERTEX SET (GFVS)</b></p> <p><b>Input:</b> A <math>\Sigma</math>-labeled graph <math>(G, \Lambda)</math> and an integer <math>k</math>.</p> <p><b>Question:</b> Does there exist a set <math>X \subseteq V(G)</math> of at most <math>k</math> vertices, such that there is no non-null cycle in <math>(G \setminus X, \Lambda)</math>?</p>	<p><b>Parameter:</b> <math>k</math></p>
--	---

As observed in [13], for a graph excluding a non-null cycle we can define a consistent labeling.

**Definition 3.** For a  $\Sigma$ -labeled graph  $(G, \Lambda)$  we call  $\lambda : V \rightarrow \Sigma$  a *consistent labeling* iff for each arc  $(u, v) = a \in A(G)$  we have  $\lambda(v) = \lambda(u) \cdot \Lambda(a)$ .

**Lemma 4** ([13]). A  $\Sigma$ -labeled graph  $(G, \Lambda)$  has a consistent labeling iff it does not contain a non-null cycle.

Note that when analyzing the complexity of the GFVS problem, it is important how the group  $\Sigma$  is represented. In [13] it is assumed that  $\Sigma$  is given via its multiplication table as a part of the input. In this paper we assume a more general model, where operations in  $\Sigma$  are computed by an oracle in polynomial time. More precisely, we assume that the oracle can multiply two elements, return an inverse of an element, provide the neutral element  $1_\Sigma$ , or check whether two elements are equal.

As noted in [17], GFVS subsumes not only the classical FVS problem, but also OCT (with  $\Sigma = \mathbb{Z}_2$ ) and MWC (with  $\Sigma$  being an arbitrary group of size not smaller than the number of terminals). We note

<sup>1</sup>In this paper, we follow the notation of Guillemot [13].

that if  $\Sigma$  is given in the oracle model, GROUP FEEDBACK VERTEX SET subsumes also EDGE SUBSET FEEDBACK VERTEX SET, which is equivalent to SUBSET FEEDBACK VERTEX SET [9].

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  $G \setminus X$  there are no cycles with at least one edge from  $S$ ?

**Lemma 5.** *Given an ESFVS instance  $(G, S, k)$ , one can in polynomial time construct an equivalent GFVS instance  $(G', \Lambda, k)$  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  $\Sigma$ -labeling  $\Lambda$  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  $\Sigma$  can be performed by a polynomial-time oracle.  $\square$

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  $\delta$ -paths theorem [6, 15]. In particular, Kawarabayashi and Wollan proved the Erdős-Pósa property for non-null cycles in highly connected graphs, generalizing a list of previous results [15].

The study of parameterized complexity of GFVS was initiated by Guillemot [13], who presented a fixed-parameter algorithm for GFVS parameterized by  $|\Sigma| + k$  running in time<sup>2</sup>  $O^*(2^{O(k \log |\Sigma|)})$ . When parameterized by  $k$ , Guillemot showed a fixed-parameter algorithm for the easier edge-deletion variant of GFVS, running in time  $O^*(2^{O(k \log k)})$ . Very recently, Kratsch and Wahlström presented a randomized kernelization algorithm that reduces the size of a GFVS instance to  $O(k^{2|\Sigma|})$  [17].

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 [13], 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 in the instance, as in Lemma 5. Hence, the usage 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 [9] and in MULTIWAY CUT [8, 19]. This suggests that the phenomenon can be, in fact, more general.

**Our result and techniques.** Our main result is a fixed-parameter algorithm for GFVS parameterized by the size of the cutset only.

**Theorem 6.** 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 [17]: 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  $|\Sigma|$ ,

<sup>2</sup>The  $O^*(\ )$  notation suppresses terms polynomial in the input size.

we carefully analyze 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 [9] 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.

Note that the bound on the running time of our algorithm matches the currently best known algorithm for SUBSET FEEDBACK VERTEX SET [9]. Therefore, we obtain the same running time as in [9] by applying a much more general framework.

In the GROUP FEEDBACK VERTEX SET problem definition in [13] a set of forbidden vertices  $F \subseteq V(G)$  is additionally given as a part of the input. Observe that one can easily gadget such vertices by replacing each forbidden vertex by a clique of size  $k + 1$  labeled with  $1_\Sigma$ ; therefore, for the sake of simplicity we assume that all the vertices are allowed.

## 2 Preliminaries

**Notation.** We use standard graph notation. For a graph  $G$ , by  $V(G)$  and  $E(G)$  we denote its vertex and edge sets, respectively. In case of a directed graph  $G$ , we denote the arc set of  $G$  by  $A(G)$ . 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  $N_G(X) = N_G[X] \setminus X$ . 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  $G \setminus X$  we denote the graph with the vertices or edges of  $X$  removed; in case of vertex removal, we remove also all the incident edges.

## 3 Algorithm

In this section we prove Theorem 6. We proceed with a standard application of the iterative compression technique in Section 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 with it. We first prepare the COMPRESSION GROUP FEEDBACK VERTEX SET instance by *untangling* it in Section 3.2, in the same manner as it is done in the kernelization algorithm of [17]. The main step of the algorithm is done in Section 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 (Section 3.4).

### 3.1 Iterative compression

The first step in the proof of Theorem 6 is a standard technique in the design of parameterized algorithms, that is, iterative compression, introduced by Reed et al. [22]. Iterative compression was also the first step of the parameterized algorithm for SUBSET FEEDBACK VERTEX SET [9].

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 with  $Z$ .

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

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

**Lemma 7.** COMPRESSION GROUP FEEDBACK VERTEX SET *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 6.

*Proof of Theorem 6.* 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 7 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  $V = \{v_1, \dots, v_n\}$ . For  $0 \leq i \leq n$  define  $V_i = \{v_1, \dots, v_i\}$  (in particular  $V_0 = \emptyset$ ) and let  $\Lambda_i$  be the function  $\Lambda$  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  $i = 1, \dots, n$  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 guess by trying all possibilities, a subset of vertices  $Z'_i \subseteq Z_i$  that is not removed in a solution of size  $k$  to  $(G[V_i], \Lambda_i)$  and use Lemma 7 for the instance  $I_{Z'_i} = (G[V_i \setminus (Z_i \setminus Z'_i)], \Lambda_i, k' = |Z'_i| - 1, Z'_i)$ . If for each set  $Z'_i$  the algorithm from Lemma 7 returns NO, then there is no solution for  $(G[V_i], \Lambda_i)$  and, consequently, there is no solution for  $(G, \Lambda)$ . However, if for some  $Z'_i$  the algorithm from Lemma 7 returns a set  $X'_i$  of size smaller than  $|Z'_i|$ , then we set  $X_i = (Z_i \setminus Z'_i) \cup X'_i$ . Since  $|X_i| = |Z_i \setminus Z'_i| + |X'_i| < |Z_i| = k + 1$ , the set  $X_i$  is a solution of size at most  $k$  for the instance  $(G_i, \Lambda_i)$ .

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 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 Section 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 7 we use the concept of *untangling*, previously used by Kratsch and Wahlström [17]. We transform an instance of C-GFVS to ensure that each arc  $(u, v)$  with both endpoints in  $V(G) \setminus Z$  is labeled  $1_\Sigma$  by  $\Lambda$ .

**Definition 8.** We call an instance  $(G = (V, A), \Lambda, k, Z)$  of C-GFVS *untangled*, iff for each arc  $(u, v) \in A$  such that  $u, v \in V \setminus Z$  we have  $\Lambda((u, v)) = 1_\Sigma$ .

Moreover, by *untangling* a labeling  $\Lambda$  around vertex  $v$  with a group element  $g$  we mean changing the labeling to  $\Lambda' : A \rightarrow \Sigma$ , 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 9.** *Let  $(G = (V, A), \Lambda)$  be a  $\Sigma$ -labeled graph,  $x \in V$  be a vertex of  $G$  and let  $g \in \Sigma$  be a group element. For any subset of vertices  $X \subseteq V$  the graph  $(G \setminus X, \Lambda)$  contains a non-null cycle iff  $(G \setminus X, \Lambda')$  contains a non-null cycle, where  $\Lambda'$  is the labeling  $\Lambda$  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  $\Lambda(C) = \Lambda'(C)$ . □

In Section 3.3 we prove the following lemma.

**Lemma 10.** **COMPRESSION GROUP FEEDBACK VERTEX SET** *for untangled instances can be solved in  $O^*(2^{O(|Z|(\log k + \log |Z|))} \cdot 2^k)$  time and polynomial space.*

Having Lemmata 9 and 10 we can prove Lemma 7.

*Proof of Lemma 7.* Let  $(G, \Lambda, k, Z)$  be an instance of C-GFVS. Since  $(G \setminus Z)$  has no non-null cycle, by Lemma 4 there is a consistent labeling  $\lambda$  of  $(G \setminus Z, \Lambda)$ .

Let  $\Lambda'$  be a result of untangling  $\Lambda$  around each vertex  $v \in V(G) \setminus Z$  with  $\lambda(v)$ . Note that, by associativity of  $\Sigma$ , 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  $u, v \in V(G) \setminus Z$ , we have  $\Lambda'(a) = (\lambda(u) \cdot \Lambda(a)) \cdot \lambda(v)^{-1} = \lambda(v) \cdot \lambda(v)^{-1} = 1_\Sigma$ . Therefore, by Lemma 9 the instance  $(G, \Lambda', k, Z)$  is an untangled instance of C-GFVS, which is a YES-instance iff  $(G, \Lambda, k, Z)$  is a YES-instance. Consequently, we can use Lemma 10 and the claim follows. □

### 3.3 Fixing a labeling on $Z$

In this section we prove Lemma 10 using the following lemma, which we prove in Section 3.4.

**Lemma 11.** *Let  $(G, \Lambda, k, Z)$  be an untangled instances of C-GFVS. There is an algorithm which for a given function  $\phi : Z \rightarrow \Sigma$ , finds a set  $X \subseteq V(G) \setminus Z$  of size at most  $k$ , such that there exists a consistent labeling  $\lambda : V(G) \setminus X \rightarrow \Sigma$  of  $(G \setminus X, \Lambda)$ , 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  $\phi$  and use the algorithm from Lemma 11. 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  $\phi$ ; together with Lemma 11 this suffices to prove Lemma 10.

**Definition 12.** Let  $(G, \Lambda, k, Z)$  be an untangled instance of C-GFVS, let  $z$  be a vertex in  $Z$  and by  $\Sigma_z$  denote the set  $\Lambda(\{(z, v) \in A(G) : v \in V(G) \setminus Z\})$ .

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

Less formally, in the flow graph we take the underlying undirected graph of  $G[V(G) \setminus Z]$  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  $V(G) \setminus Z$  labeled with  $g$  by  $\Lambda$ . A vertex  $g \in \Sigma_z$  is adjacent to all the vertices of  $V(G) \setminus Z$  for which there exists an arc going from  $z$ , labeled with  $g$  by  $\Lambda$ .

**Lemma 13.** *Let  $(G, \Lambda, k, Z)$  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  $v \in V(G) \setminus Z$ , in  $H$  there are at least  $k + 2$  paths from  $v$  to  $\Sigma_z$  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  $X \subseteq V(G) \setminus Z$ , where  $|X| \leq k$ . Then there at least 2 out of the  $k + 2$  paths from  $v$  to  $\Sigma_z$  remain in  $H \setminus X$ . These two paths are vertex disjoint apart from  $v$ , so they correspond to a non-null cycle in  $G \setminus X$ , a contradiction.  $\square$

**Definition 14.** For an untangled instance  $(G, \Lambda, k, Z)$  of C-GFVS by an *external path* we denote any path  $P$  beginning and ending in  $Z$ , but with all internal vertices belonging to  $V(G) \setminus Z$ . 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  $\Lambda(P) = g$ .

**Lemma 15.** *Let  $(G, \Lambda, k, Z)$  be an untangled instance of C-GFVS. If for each  $z \in Z$  and  $v \in V(G) \setminus Z$  there are at most  $k + 1$  vertex disjoint paths from  $v$  to  $\Sigma_z$  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  $(G, \Lambda, k, Z)$ .*

*Proof.* Let us assume that  $X \subseteq V(G) \setminus Z$  is a solution for  $(G, \Lambda, k, Z)$ . 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  $v \in X$ , 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 with  $X$ , creating a non-null cycle disjoint with  $X$ . This cycle is not necessarily simple; however, if it is non-null, then it contains a simple non-null subcycle that is also disjoint with  $X$ .

Consider a connected component  $C$  of  $G[V(G) \setminus Z]$  to which  $v$  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  $\Lambda$ , 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  $(G \setminus X, \Lambda)$ , we infer that in  $H_0 = H[C \cup T] \setminus (X \cap C)$ , 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  $v$  and  $t$  in  $H[C \cup T]$ . Let  $w_t$  be the closest to  $t$  vertex from  $X$  on the path  $P_t$ . As  $|X| \leq k$  and  $|T| > k(k + 1)$ , there exists  $w \in X$  such that  $w = w_t$  for at least  $k + 2$  elements  $t \in T$ . By the definition of the vertices  $w_t$  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  $w_t$  for all  $t$  with  $w = w_t$  are vertex disjoint apart from  $w$ . As there are at least  $k + 2$  of them, we have a contradiction.  $\square$

We are now ready to prove Lemma 10 given Lemma 11.

*Proof of Lemma 10.* If there exists a vertex  $v$ , satisfying the properties of Lemma 13, 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  $v \in V(G) \setminus Z$ , there are at most  $k + 1$  vertex disjoint paths from  $v$  to  $\Sigma_z$  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.

By Lemma 15, 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. Observe, that one can easily verify the cardinality of  $\Sigma(z_1, z_2)$ , 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  $G[V(G) \setminus Z]$ . Clearly, this can be done in polynomial time.

Knowing that the sets  $\Sigma(z_1, z_2)$  have sizes bounded by a function of  $k$ , we can enumerate all the reasonable labelings of  $Z$ . 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  $G \setminus X$ , for some fixed solution  $X \subseteq V(G) \setminus Z$ . Let  $F$  be any spanning forest of  $G'$ . Since  $F$  has at most  $|Z| - 1$  edges, we can guess  $F$ , by trying at most  $|Z| \cdot |Z|^{2(|Z|-1)}$  possibilities. 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 fixed vertex from each connected component of  $F$  with  $1_\Sigma$ . For all other vertices of  $F$  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  $\phi^{-1}(z_1) \cdot \phi(z_2) \in \Sigma(z_1, z_2)$ . Hence, we can exhaustively try  $2^{O(|Z|(\log k + \log |Z|))}$  labelings  $\phi$  of  $Z$ , and use Lemma 11 for each of them.  $\square$

### 3.4 Reduction to Multiway Cut

In this section, we prove Lemma 11, 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  $|\Sigma|$  [17]. Currently the fastest FPT algorithm for MULTIWAY CUT is due to Cygan et al. [8], 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 11.* Firstly, we check whether the given function  $\phi$  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  $\Sigma$ -labeled graph  $(G, \Lambda)$ , a set  $Z$ , an integer  $k$ , and a function  $\phi : Z \rightarrow \Sigma$ , 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 in the set  $T$  there exactly these elements of  $\Sigma$ , which are potential values of a consistent labeling of  $(G, \Lambda)$  that matches  $\phi$  on  $Z$ . As the edge set, we set  $E = \{uv : (u, v) \in A(G[V(G) \setminus Z])\} \cup \{gv : (u, v) \in A(G), u \in Z, v \in V(G) \setminus Z, \phi(u) \cdot \Lambda((u, v)) = g\}$ . We show that  $(G', T, k)$  is a YES-instance of MULTIWAY CUT iff there exists a set  $X \subseteq V(G) \setminus Z$ , such that there exists a consistent labeling  $\lambda$  of  $(G \setminus X, \Lambda)$  with  $\lambda|_Z = \phi$ .

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

Let  $X \subseteq V(G) \setminus Z$  be a set of vertices of  $G$ ,  $|X| \leq k$ , such that there is a consistent labeling  $\lambda$  of  $(G \setminus X, \Lambda)$ , where  $\lambda|_Z = \phi$ . By the definition of edges between  $T$  and  $V(G) \setminus Z$  in  $G'$ , each vertex of

$V(G) \setminus Z$  is reachable from at most one terminal in  $G'$ , since otherwise  $\lambda$  would not be a consistent labeling of  $(G \setminus X, \lambda)$ . Therefore,  $X$  is a solution for  $(G', T, k)$ .

We can now apply the algorithm for MULTIWAY CUT of [8] to the instance  $(G', T, k)$  in order to conclude the proof.  $\square$

## 4 Conclusions and open problems

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 oracle model, that allows us to generalize the recent algorithm for SUBSET FEEDBACK VERTEX SET [9] within the same complexity bound.

We would like to note that if we represent group elements by strings consisting  $g$  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 oracle.

Both our algorithm and the algorithm for SUBSET FEEDBACK VERTEX SET of [9] seems hard to speed up to time complexity  $O^*(2^{O(k)})$ . Can these problems be solved in  $O^*(2^{O(k)})$  time, or can we prove that such a result would violate Exponential Time Hypothesis?

**Acknowledgements.** We thank Stefan Kratsch and Magnus Wahlström for inspiring discussions on graph separation problems and for drawing our attention to the GROUP FEEDBACK VERTEX SET problem.

## References

- [1] Ann Becker, Reuven Bar-Yehuda, and Dan Geiger. Randomized algorithms for the loop cutset problem. *J. Artif. Intell. Res. (JAIR)*, 12:219–234, 2000.
- [2] Hans L. Bodlaender. On disjoint cycles. *Int. J. Found. Comput. Sci.*, 5(1):59–68, 1994.
- [3] Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set new measure and new structures. In Haim Kaplan, editor, *SWAT*, volume 6139 of *Lecture Notes in Computer Science*, pages 93–104. Springer, 2010.
- [4] Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. Syst. Sci.*, 74(7):1188–1198, 2008.
- [5] Jianer Chen, Yang Liu, Songjian Lu, Barry O’Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.
- [6] Maria Chudnovsky, Jim Geelen, Bert Gerards, Luis A. Goddyn, Michael Lohman, and Paul D. Seymour. Packing non-zero  $a$ -paths in group-labelled graphs. *Combinatorica*, 26(5):521–532, 2006.
- [7] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *52th Annual IEEE Symposium on Foundations of Computer Science (FOCS) (to appear)*, 2011.
- [8] Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. In *IPEC (to appear)*, 2011. Available at <http://arxiv.org/abs/1107.1585>.
- [9] Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 449–461. Springer, 2011.

- [10] Frank K. H. A. Dehne, Michael R. Fellows, Michael A. Langston, Frances A. Rosamond, and Kim Stevens. An  $O(2^{O(k)})n^3$  fpt algorithm for the undirected feedback vertex set problem. *Theory Comput. Syst.*, 41(3):479–492, 2007.
- [11] Rodney G. Downey and Michael R. Fellows. Fixed parameter tractability and completeness. In *Complexity Theory: Current Research*, pages 191–225, 1992.
- [12] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer, 1999.
- [13] Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61–71, 2011.
- [14] Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, and Sebastian Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *J. Comput. Syst. Sci.*, 72(8):1386–1396, 2006.
- [15] Ken ichi Kawarabayashi and Paul Wollan. Non-zero disjoint cycles in highly connected group labelled graphs. *J. Comb. Theory, Ser. B*, 96(2):296–301, 2006.
- [16] Iyad A. Kanj, Michael J. Pelsmajer, and Marcus Schaefer. Parameterized algorithms for feedback vertex set. In Rodney G. Downey, Michael R. Fellows, and Frank K. H. A. Dehne, editors, *IWPEC*, volume 3162 of *Lecture Notes in Computer Science*, pages 235–247. Springer, 2004.
- [17] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *CoRR*, abs/1111.2195, 2011.
- [18] Dániel Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.
- [19] N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. LP can be a cure for parameterized problems. In *Proc. of STACS'12 (to appear)*, 2012.
- [20] Venkatesh Raman, Saket Saurabh, and C. R. Subramanian. Faster fixed parameter tractable algorithms for finding feedback vertex sets. *ACM Transactions on Algorithms*, 2(3):403–415, 2006.
- [21] Igor Razgon and Barry O’Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.
- [22] Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.
- [23] Stéphan Thomassé. A  $4k^2$  kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.