A Faster Parameterized Algorithm for Group Feedback Edge Set

M.S. Ramanujan(✉)

Vienna University of Technology, Vienna, Austria
ramanujan@ac.tuwien.ac.at

Abstract. In the Group Feedback Edge Set (ℓ) (Group FES(ℓ)) problem, the input is a group-labeled graph G over a group Γ of order ℓ and an integer k and the objective is to test whether there exists a set of at most k edges intersecting every non-null cycle in G. The study of the parameterized complexity of Group FES(ℓ) was motivated by the fact that it generalizes the classical Edge Bipartization problem when $\ell = 2$. Guillemot [IWPEC 2008, Discrete Optimization 2011] initiated the study of the parameterized complexity of this problem and proved that it is fixed-parameter tractable (FPT) parameterized by k. Subsequently, Wahlström [SODA 2014] and Iwata et al. [2014] presented algorithms running in time $\mathcal{O}(4^k n^{\mathcal{O}(1)})$ (even in the oracle access model) and $\mathcal{O}(\ell^{2k} m)$ respectively. In this paper, we give an algorithm for Group FES(ℓ) running in time $\mathcal{O}(4^k k^3 \ell(m + n))$. Our algorithm matches that of Iwata et al. when $\ell = 2$ (upto a multiplicative factor of k^3) and gives an improvement for $\ell > 2$.

1 Introduction

In a covering problem we are given a universe of elements U, a family $\mathcal{F}$ ($\mathcal{F}$ could be given implicitly) and an integer k and the objective is to check whether there exists a subset of U of size at most k which intersects all the elements of $\mathcal{F}$. Several natural problems on graphs can be framed as a covering problem. One of the most well-studied covering problems are the *feedback set* problems. In these problems, the family $\mathcal{F}$ is a succinctly defined subset of the set of cycles in the given graph. For instance, in the Feedback Vertex Set problem, the objective is to decide whether there exists a vertex subset S (also called a transversal) of size at most k which intersects *all* cycles in the graph. That is, $\mathcal{F}$ is the set of all cycles in the input graph. In the classical Odd Cycle Transversal (Edge Bipartization) problem, the objective is to decide whether there exists a vertex subset (respectively edge subset) S of size at most k which intersects all *odd* cycles.

Yet another kind of feedback set problem deals with *gain-graphs* or *group-labeled graphs* and is called Group Feedback Vertex Set. Group-labeled graphs are generalizations of the well-studied class of *signed-graphs* introduced by Harary [11]. These are directed graphs where the arcs are labeled by elements of a group Γ and whenever multiplying the arc labels in order around a cycle

P. Heggernes (Ed.): WG 2016, LNCS 9941, pp. 269–281, 2016.
DOI: 10.1007/978-3-662-53536-3_23

results in an element other than 1_Γ – the identity element of the group, the cycle under consideration is called a *non-null* cycle.

In the Group Feedback Edge Set (Group FES) problem, the objective is to check whether there is a set of at most k arcs that intersect all non-null cycles in a given Γ-labeled graph for some finite group Γ. In the Group FES(ℓ) problem, we require Γ to have order ℓ. Similarly, in the Group FVS problem, the objective is to hit all non-null cycles with at most k vertices and in Group FVS(ℓ), the group is required to have order ℓ. Although group-labeled graphs and the Group FVS problem has been studied from a graph-theoretic point of view (see for example [9,14,21]) the study of the parameterized complexity of both versions of this problem was first initiated by Guillemot [10]. Formally, a *parameterization* of a problem is the assignment of an integer k to each input instance and we say that a parameterized problem is FPT if there is an algorithm that solves the problem in time $f(k) \cdot |I|^{\mathcal{O}(1)}$, where $|I|$ is the size of the input instance and f is an arbitrary computable function depending only on the parameter k. For more background, the reader is referred to the books [3,7,8,18]. Guillemot [10] showed that Group FVS(ℓ) is FPT parameterized by k and ℓ and Group FES(ℓ) is FPT parameterized by k by giving algorithms running in time $\mathcal{O}^*((4\ell+1)^k)$ and $\mathcal{O}^*((8k+1)^k)$ respectively (the $\mathcal{O}^*()$ notation subsumes polynomial factors).

The first single-exponential FPT algorithm for Group FES was given by Wahlström [20] who extended the branching algorithm of Guillemot to a more sophisticated LP-guided branching algorithm based on newly developed tools from the theory of valued constraint satisfaction. We remark that this algorithm, which runs in time $\mathcal{O}^*(4^k)$ was designed for the more general vertex version of the problem and improved upon the work of Cygan et al. [4] who obtained the first FPT algorithm for Group FVS parameterized only by k. In fact, the algorithm of Wahlström as well as that of Cygan et al. works in the oracle access model where the group is not given via its multiplication table but in the form of a polynomial time oracle. However, this algorithm relies on solving linear programs and hence has a dependence on the input-size that is far from linear even when the group-size ℓ is constant. A recently studied generalization of Group FES(ℓ) is the Unique Label Cover(ℓ) problem where the input is a graph labeled by permutations of $[\ell]$, that is, elements of the symmetric group S_ℓ and the objective is to delete at most k edges such that the resulting graph can be labeled by elements of $[\ell]$ in a way that 'respects' the permutations of $[\ell]$ on the arcs. The fact that Unique Label Cover(ℓ) generalizes Group FES(ℓ) follows from Cayley's Theorem which states that every finite group of order ℓ is isomorphic to a subgroup of S_ℓ.

Chitnis et al. [2] were the first to prove that Unique Label Cover(ℓ) is FPT parameterized by ℓ and k. In fact they showed that under standard complexity hypotheses, to obtain fixed-parameter tractability, parameterizing by both ℓ and k is unavoidable. Subsequently, Wahlström [20] improved upon this result by giving an algorithm that runs in time $\mathcal{O}^*(\ell^{2k})$. Following this work, Iwata et al. [13] gave an algorithm for Unique Label Cover(ℓ) that runs in time $\mathcal{O}(\ell^{2k}m)$

where m is the number of edges in the input graph. This was the first linear-time FPT algorithm for this problem and implies an algorithm for Group FES(ℓ) running in time $\mathcal{O}(\ell^{2k}m)$. Hence, prior to this work, the $\mathcal{O}^*(4^k)$ algorithm in [20] and the $\mathcal{O}(\ell^{2k}m)$ algorithm in [13] were the best known FPT algorithms for Group FES(ℓ) with respect to dependence on the parameter and the input and group sizes respectively. In this paper, we give an algorithm for Group FES(ℓ) that comes close to matching the best of both algorithms. We obtain an algorithm that has a dependence of $4^{k+\mathcal{O}(\log k)}$ on k and a dependence of $\mathcal{O}(\ell(m+n))$ on the input and group-sizes. In fact this algorithm outperforms that of Iwata et al. [13] for all $\ell > 2$. We now give a formal description of the problem under consideration and state our result.

Group FES(ℓ) **Parameter:** k
Input: A Γ-labeled graph (G, Λ) where $|\Gamma| = \ell \geq 2$, integer k.
Question: Is there a set $X \subseteq A(G)$ of size at most k such that $G - X$ has no non-null cycles?

Theorem 1. Group FES*(ℓ) can be solved in time* $\mathcal{O}(4^k k^3 \ell(m+n))$ *where* m *and* n *denote the number of arcs and vertices in the input graph respectively.*

Methodology. We closely follow the template developed for solving graph separation problems via important separators in [1,16], those via LP-guided branching in [5,10,12,15], the Valued CSP-based algorithms in [13,20] and the skew-symmetric branching algorithm for 2-SAT Deletion in [19]. The common thread connecting these algorithms is that they all begin by proving a 'persistence lemma' or Nemhauser-Trotter-type theorem. In these lemmas, one proves that the solution to an appropriate linear program or a maximum-flow question on an appropriate network can be used to 'fix a configuration' for vertices which satisfy certain properties. For instance, the classical Nemhauser-Trotter Theorem [17] for Vertex Cover states that if an optimal solution to the standard relaxation of the Vertex Cover Integer Linear Program (ILP) assigns 0 or 1 to a vertex then there is also an optimal solution to the ILP which does the same with respect to this vertex.

However, in this work the persistence lemma we prove will be based directly on the solution to a max-flow question in a network as opposed to using the solution to a linear program. The reason behind this is that the structural properties of Group FES(ℓ) closely resemble those of the classical Edge Multiway Cut problem while the vertex version, Group FVS(ℓ) is closely related to the Node Multiway Cut problem (see [4,10]). As a result, we are able to design our persistence lemma using an appropriate analogue of the classical notion of 'isolating cuts' from [6]. Once we prove this lemma, we design a natural reduction rule based on it and describe a subroutine that runs in polynomial time (with a linear dependence on ℓ and $m+n$) and either finds a valid application of the reduction rule or an arc on which a naive branching step will decrease a predetermined measure for the input. Finally, we remark that while the structure of our algorithm strongly resembles that of the algorithms in the works cited above, the fact that we are dealing with groups of order greater than 2 while trying to simultaneously optimize the dependence of the running time in 'three-dimensions' – parameter, input-size and group-size, poses non-trivial obstacles when it comes to actually implementing each step.

2 Preliminaries

Let Γ be a group with identity element 1_Γ. A Γ-labeled graph is a pair (G, Λ) where G is a digraph with at most one arc between every pair of vertices and $\Lambda : A(G) \rightarrow \Gamma$. If $(u, v) \in A(G)$, then we denote by $\Lambda(v, u)$ the group element $\Lambda(u, v)^{-1}$. For a digraph G, we denote by $\tilde{G}$ the underlying undirected graph. Let $P = v_1, \ldots, v_\ell$ be a path in $\tilde{G}$. We denote by $\Lambda(P)$ the group element $\Lambda(v_1, v_2) \cdot \Lambda(v_2, v_3) \cdots \Lambda(v_{\ell-1}, v_\ell)$. Let $C = v_1, \ldots, v_\ell, v_1$ be a cycle in $\tilde{G}$. We denote by $\Lambda(C)$ the group element $\Lambda(v_1, v_2) \cdot \Lambda(v_2, v_3) \cdot \Lambda(v_{\ell-1}, v_\ell) \cdots \Lambda(v_\ell, v_1)$. We call C *non-null* if $\Lambda(C) \neq 1_\Gamma$. Note that even though different choices of the vertex v_1 in the same cycle may lead to different values for $\Lambda(C)$, it is easy to see that if for one choice of v_1 the value of $\Lambda(C)$ is not 1_Γ then for no choice of v_1 is it 1_Γ.

For an undirected graph H and vertex set $Z \subseteq V(H)$, we denote by $\delta(Z)$ the set of edges which have exactly one endpoint in Z. We denote by $E(Z)$ the edges of H which have both endpoints in Z. This notation also extends to directed graphs as $A(Z)$. For a set X of edges in an undirected graph or arcs in a directed graph, we denote by $V(X)$ the set of endpoints of the edges or arcs in X. For a vertex subset X, $N(X)$ denotes the set of neighbors of X and $N[X]$ denotes the set $X \cup N(X)$. For an undirected graph G and disjoint vertex sets X and Y, a path is called an X-Y path if it has one endpoint in X and the other in Y and a set $S \subseteq E(G)$ is said to be an X-Y separator if there is no X-Y path in the graph $G - S$. We denote the vertices in the components of $G - S$ which intersect X by $R(X, S)$. We denote by $\lambda_G(X, Y)$ the size of the smallest X-Y separator in G. Due to space constraints, proofs of Lemmas marked [$\star$] have been omitted from the extended abstract and can be found in the full version of the paper.

3 Consistent Labelings and the Auxiliary Graph

In this section, we begin by recalling known results on group-labeled graphs that exclude a non-null cycle. Following that, we will associate an auxiliary graph with every instance of Group FES(ℓ).

Definition 1. *Let (G, Λ) be a Γ-labeled graph and let $\Psi : V(G) \rightarrow \Gamma$. We say that Ψ is a* ***consistent labeling*** *for this graph if for all $(u, v) = a \in A(G)$, $\Psi(u) \cdot \Lambda(a) = \Psi(v)$.*

Lemma 1 [10]**.** *Let (G, Λ) be a Γ-labeled graph. There is no non-null cycle in G if and only if G has a consistent labeling.*

Observation 2. *Let (G, Λ) be a Γ-labeled graph. If G has a consistent labeling, then for every $v \in V(G)$ and $g \in \Gamma$, there is a consistent labeling ψ_g^v for G such that $\psi_g^v(v) = g$.*

Definition 2. *Let (G, Λ) be a Γ-labeled graph and let $\Psi : V(G) \to \Gamma$ be a consistent labeling for G. For a set $Z \subseteq V(G)$ and a function $\tau : Z \to \Gamma$, we say that Ψ* ***agrees with*** *τ on Z if for every $v \in Z$, $\Psi(v) = \tau(v)$.*

We consider a slightly more general formulation of the Group FES(ℓ) problem, where the input contains (G, Λ, k), a set $\hat{Z}$ such that $G[\hat{Z}]$ is connected and a function $\tau : \hat{Z} \to \Gamma$ such that τ *is a consistent labeling for* $G[\hat{Z}]$ and the objective is to find a solution *given that* if there is a solution then there is one whose deletion leaves a graph which has a consistent labeling that *agrees* with τ on $\hat{Z}$. That is, we may assume that we are looking for a solution whose deletion allows a consistent labeling that 'extends' τ. Clearly this formulation is more general since we can simply set $\hat{Z} = \emptyset$ to begin with and leave τ undefined. For a given $\hat{Z}$ and $\tau : \hat{Z} \to \Gamma$, we denote by $\hat{Z}_\tau$ the set $\{z_\alpha | z \in \hat{Z}, \alpha = \tau(z)\}$. We now define the auxiliary graph associated with the instance. We will be performing almost all of our computations in this graph.

The Auxiliary Graph and Some Properties. For an instance $I = (G, \Lambda, k, \hat{Z}, \tau)$ of Group FES(ℓ), we define an associated auxiliary graph H_I as follows. The vertex set of H_I is $\{v_g | v \in V(G), g \in \Gamma\}$. The vertex v_g represents the existence of an (eventual) consistent labeling of G where v is assigned the group element g. The edge set of H_I is defined as follows. For every arc $a = (u, v) \in A(G)$ and for every $g \in \Gamma$, there is an edge $(u_g, v_{g \cdot \Lambda(a)})$. Observe that corresponding to a, there are exactly ℓ edges in H_I and furthermore, these form a matching. Therefore, H_I has ℓn vertices and ℓm edges, where n and m are the number of vertices and edges in G respectively. Note that the graph H_I in fact only depends on (G, Λ). However, we choose to denote the graph as H_I in order to facilitate an easier presentation in the description of the algorithm. Moving forward, we will characterize the dependencies between vertices when subjected to certain constraints. Before we do so, we need the following definitions and observation.

Definition 3. *Let $I = (G, \Lambda, k, \tau)$ be an instance of* Group FES*(ℓ). For $v \in V(G)$, we use $[v]$ to denote the set $\{v_g | g \in \Gamma\}$. For a subset $S \subseteq V(G)$, we use $[S]$ to denote the set $\bigcup_{v \in S}[v]$. Similarly, for an arc $a = (u, v) \in A(G)$, we use $[a]$ to denote the set $\{(u_i, v_j)\}_{i \in \Gamma, j = \Lambda(a) \cdot i}$ of edges in H_I and for a subset $X \subseteq A(G)$, we use $[X]$ to denote the set $\bigcup_{a \in X}[a]$. For the sake of convenience, we also reuse the same notation in the following way. For every $v \in V(G)$ and $\alpha \in \Gamma$, we denote by $[v_\alpha]$ the set $[v]$. Similarly, for every $a = (u, v) \in A(G)$ and $\alpha, \beta \in \Gamma$ such that $e = (u_\alpha, v_\beta) \in E(H_I)$, we denote by $[e]$ the set $[a]$. This definition extends in a natural way to sets of vertices and edges of the auxiliary graph H_I. For a set $S \subseteq V(H_I) \cup E(H_I)$, we denote by S^{-1} the set $\{s | s \in V(G) \cup A(G) : [s] \cap S \neq \emptyset\}$. For an arc $a \in A(G)$ and edge $e \in [a]$, we also use e^{-1} to denote the arc a.*

Observation 3 *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* Group FES*(ℓ). Then the following statements hold. (a) For every $v \in V(G)$, for every distinct $g_1, g_2 \in \Gamma$, v_{g_1} and v_{g_2} have no common neighbors in H_I. (b) For a set $S \subseteq A(G)$, $H_I - [S] = H_{I'}$ where $I' = (G - S, \Lambda, 0, \hat{Z}, \tau)$. (c) If Ψ is a consistent labeling for G, then for any $u, v \in V(G)$, if u_g is in the same connected component as $v_{g'}$ in H_I where $g = \psi(v)$ then $\Psi(u) = g'$.*

Definition 4. *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* Group FES*(ℓ). We say that a set $Z \subseteq V(H_I) \cup E(H_I)$ is* ***regular*** *if $|Z \cap [v]| \leq 1$ for any $v \in V(G)$ and $|Z \cap [a]| \leq 1$ for any $a \in A(G)$. We say that Z is* ***irregular*** *otherwise. That is, regular sets contain at most 1 copy of any vertex and arc of G.*

Now that we have defined the notion of regularity of sets, we prove the following lemma which shows that the auxiliary graph displays a certain symmetry with respect to regular paths. This will allow us to transfer arguments which involve a regular path between vertices v_{g_1} and u_{g_2} to one between vertices v_{g_3} and u_{g_4} where $g_1 \neq g_3$ and $g_2 \neq g_4$.

Lemma 2 [$\star$]**.** *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* Group FES*(ℓ). Let P be a regular path in H_I from v_g to $u_{g'}$ for some $u, v \in V(G)$ and $g, g' \in \Gamma$. Let $V(P)$ denote the set of vertices of G in P and let U denote the set $[V(P)]$. Then, there is a set $\mathcal{P} = \{P_r\}_{r \in \Gamma}$ of ℓ vertex disjoint regular paths in H_I and a partition $\mathcal{U}$ of U into sets $\{U_r\}_{r \in \Gamma}$ such that for each $\gamma \in \Gamma$, $V(P_\gamma) = U_\gamma$ and P_γ is a path from v_γ to $u_{\gamma \cdot \Lambda(P^{-1})}$.*

Observation 4 *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* Group FES*(ℓ). Then, the following statements hold. (a) The set $\hat{Z}_\tau$ is regular and furthemore, $H_I[\hat{Z}_\tau]$ is connected. (b) If $S \subseteq A(G)$ is such that $[S]$ intersects all $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ paths in H_I then $R(\hat{Z}_\tau, [S])$ is regular. (c) If $S \subseteq A(G)$ is a minimal set such that $G - S$ has a consistent labeling that agrees with τ on $\hat{Z}$, then $[S]$ is disjoint from $A(\hat{Z})$.*

Using the observations and structural lemmas proved so far, we will now give a forbidden-structure characterization of 'solved' Yes instances of Group FES(ℓ), that is instances where $k = 0$.

Lemma 3 [$\star$]**.** *Let $I = (G, \Lambda, 0, \hat{Z}, \tau)$ be a* Yes *instance of* Group FES*(ℓ) where G is connected. Let $v \in V(G)$ and $g \in \Gamma$. Then, there is a consistent labeling Ψ such that $\Psi(v) = g$ if and only if there is no $g' \in \Gamma$ such that v_g and $v_{g'}$ are in the same connected component of H_I.*

In the next lemma, we extend the statement of the previous lemma to include a description of general Yes instances of the Group FES(ℓ) problem.

Lemma 4. *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* Group FES*(ℓ). Then, I is a* Yes *instance if and only if there is a set $S \subseteq A(G)$ of size at most k such that for every vertex $v \in V(G)$, there is no path in $H_I - [S]$ from v_g to $v_{g'}$ for any $g' \neq g$. Furthermore, if $G - S$ has a consistent labeling that agrees with τ on $\hat{Z}$ then, $[S]$ intersects all $\hat{Z}_\tau - ([\hat{Z}] \setminus \hat{Z}_\tau)$ paths in H_I.*

Proof. We first argue both directions in the first part of the statement. In the forward direction, suppose that I is a YES instance and let $S \subseteq A(G)$ be a solution. That is, $G - S$ has a consistent labeling. Now, suppose that for some distinct $g, g' \in \Gamma$, there is a path in H_I from v_g to $v_{g'}$ disjoint from $[S]$. But by Observation 3(b), this path exists in $H_{I'}$ where $I' = (G - S, \Lambda, 0, \hat{Z}, \tau)$. However, since $G - S$ has a consistent labeling, this contradicts Lemma 3, completing the argument in the forward direction.

In the converse direction, suppose that $S \subseteq A(G)$ such that $[S]$ intersects all paths from v_g to $v_{g'}$ for every $v \in V(G)$ and $g \neq g'$ in the graph H_I. Let $I' = (G - S, \Lambda, 0, \hat{Z}, \tau)$. By Observation 3(b), we know that there are no v_g-$v_{g'}$ paths in $H_{I'}$ for any $v \in V(G)$ and distinct $g, g' \in \Gamma$. But applying Lemma 3 on each connected component of $G - S$ (since the premise of this lemma requires connectivity of the graph), we conclude that $G - S$ has a consistent labeling. This completes the argument in the converse direction.

For the second statement, suppose that Ψ is a consistent labeling of G that agrees with τ on $\hat{Z}$. Suppose that for some $u, v \in \hat{Z}$ and $g, g' \in \Gamma$ where $u_g \in \hat{Z}_\tau$ and $v_{g'} \notin \hat{Z}_\tau$, v_g is in the same component of $H_I - [S]$ as $u_{g'}$. Then, Observation 3(c) implies that $\Psi(u) = g'$, a contradiction to our assumption that Ψ agrees with τ on $\hat{Z}$. This completes the proof of the lemma. □

Using the above lemma, we will interpret the GROUP FES(ℓ) problem as a parameterized cut-problem. Furthermore, observe that due to this lemma, the size of a minimum $\hat{Z}_\tau - [Z] \setminus \hat{Z}_\tau$ separator in H_I is a natural lower bound on the number of edges of H_I which correspond to the arcs in a solution for the given instance. Note that although for a solution $S \subseteq A(G)$, a naive upper bound on the number of edges in $[S]$ that are required to hit all $\hat{Z}_\tau - [Z] \setminus \hat{Z}_\tau$ paths in H_I is ℓk, we will prove shortly that the actual bound is much tighter and is in fact, independent of the group-size. This is a crucial difference between the vertex and edge variants of this problem as a similar property does not exist in the vertex version. We conclude this subsection by stating the following consequence of Lemma 4 and Observation 4(b).

Lemma 5 [⋆]. *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* GROUP FES*(ℓ) and let $S \subseteq A(G)$ be a solution for this instance such that $G - S$ has a consistent labeling that agrees with τ on $\hat{Z}$. Then, $R(\hat{Z}_\tau, [S])$ is regular.*

Closed and Open Edges. Here, we will examine the set of edges crossing a regular set and divide them into 'closed' and 'open' edges. These notions are crucial for the description of our reduction rule. Intuitively, an edge e crossing a regular set Z in H_I is considered open if its image in G, e^{-1} also crosses Z^{-1} in G. Otherwise, it is considered closed. We now define these notions in a way that is most convenient for us to invoke in our proofs.

Definition 5. *Let $Z \subseteq V(H_I)$ be a regular set in H_I and let $u, v \in V(G)$ and $\alpha, \beta \in \Gamma$ be such that $e = (u_\alpha, v_\beta) \in \delta(Z)$. We call e a* **closed edge** *in H_I with respect to Z if $Z \cap [u]$ and $Z \cap [v]$ are both non-empty. Otherwise, we say that e is an* **open edge** *in H_I with respect to Z.*

Observation 5. *Let $Z \subseteq V(H_I)$ be a regular set in H_I and let $u, v \in V(G)$ and $\alpha, \beta \in \Gamma$ be such that $e = (u_\alpha, v_\beta) \in \delta(Z)$. Then, the following statements hold.*

(a) If e is closed with respect to Z then e^{-1} has both endpoints in Z^{-1}.
(b) If e is closed with respect to Z then H_I has no edge between $Z \cap [u]$ and $Z \cap [v]$.
(c) If e is open with respect to Z then, $e^{-1} \in \delta_{\tilde{G}}(Z^{-1})$.

We now state and prove a crucial fact regarding the *number* of open and closed edges crossing any regular set in H_I.

Lemma 6 [⋆]. *Let $I = (G, \Lambda, \hat{Z}, k, \tau)$ be an instance of* Group FES*(ℓ) and let $Y \subseteq V(H_I)$ be a regular set. Then, for every edge $e \in \delta(Y)$, $|\delta(Y) \cap [e^{-1}]| = 1$ if e is open with respect to Y and $|\delta(Y) \cap [e^{-1}]| = 2$ otherwise.*

Figure 1 illustrates the structure of the edges crossing a regular set, guaranteed by Lemma 6. As a consequence of Lemma 6 and Observation 4(b), we have the following.

Lemma 7. *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* Group FES*(ℓ) and let $S \subseteq A(G)$ be a set of size at most k such that $[S]$ intersects all $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ paths in H_I. Then, $|\delta(R(\hat{Z}_\tau, [S]))| \leq 2k$.*

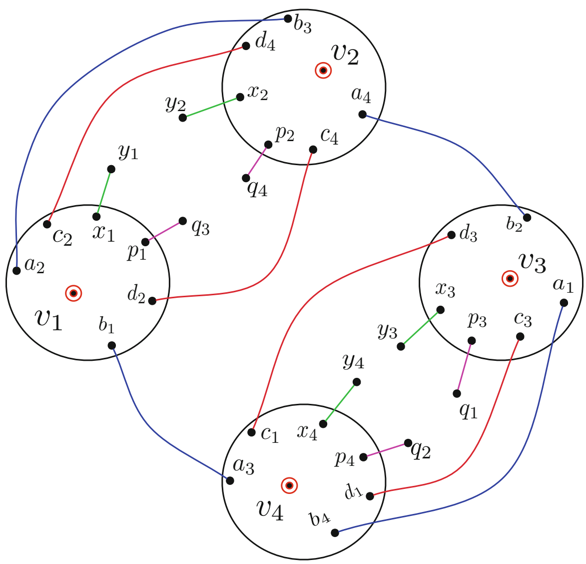

Fig. 1. An illustration of all the edges crossing the regular set Z. The four sets in the figure denote the four regular 'copies' of Z^{-1} in the graph H_I. Here, the edges $(a_2, b_3), (a_3, b_1), (c_2, d_4), (c_4, d_2)$ are closed and the edges $(x_1, y_1), (p_1, q_3)$ are open with respect to Z which is the set containing v_1

Lemma 7 implies that even though $[S]$ can contain up to ℓk edges, at most $2k$ of these are required to separate $\hat{Z}_\tau$ from $[\hat{Z}] \setminus \hat{Z}_\tau$. Motivated by this fact, we define the following measure on instances of Group FES(ℓ) which captures the gap between the budget k which is the upper bound on the size of the solution S and the size of the smallest $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator in H_I which is a lower bound on $2|S|$. Lemma 7 implies that for a Yes instance I, the gap $\mu(I)$ is always non-negative. Furthermore, it will be easy to see that for any given instance I, we can check whether $\mu(I)$ is non-negative in time $\mathcal{O}(k\ell(m+n))$.

Definition 6. *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* Group FES*(ℓ). The gap $\mu(I)$ is defined as $\mu(I) = k - \frac{1}{2}(\lambda_{H_I}(\hat{Z}_\tau, [Z] \setminus \hat{Z}_\tau))$.*

Proof of Persistence and Description of the Reduction Rule. We are now ready to prove the *Persistence* Lemma which plays a major role in the design of the algorithm. In essence this lemma says that if we find a minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator S such that $R(\hat{Z}_\tau, S)$ is regular, then we can *correctly fix* the labels of all vertices which have exactly one copy in $R(\hat{Z}_\tau, S)$ and look for a solution whose deletion allows a consistent labeling that is an extension of this one. It will be shown later that once we fix the labels of these vertices, the subsequent exhaustive branching steps will decrease the gap $\mu(I)$ by at least $\frac{1}{2}$ and since $\mu(I)$ is always required to be non-negative, the *depth* of the search tree will be bounded by $2\mu(I)$.

Lemma 8 [$\star$](**Persistence Lemma**). *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be a* Yes *instance of* Group FES*(ℓ). Let $X \subseteq A(G)$ be a minimal set of size at most k such that $G - X$ has a consistent labeling Ψ agreeing with τ on $\hat{Z}$. Let T denote the set $[\hat{Z}] \setminus \hat{Z}_\tau$. Let S be a minimum $\hat{Z}_\tau - T$ separator in H_I and let $Z = R(\hat{Z}_\tau, [S])$. Then, there is a solution for the given instance disjoint from $A(Z^{-1})$.*

Having proved the Persistence Lemma, we proceed to describe the reduction rule based on it. Before we do so, we need to prove the following lemmas.

Lemma 9 [$\star$]**.** *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be a* Yes *instance of* Group FES*(ℓ) and let S be a minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator in H_I and let $Y = R(\hat{Z}_\tau, [S])$. Then, $\delta(Y)$ is also a minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator in H_I. Furthermore, if S is not regular, then there is an edge $e \in \delta(Y)$ which is closed with respect to Y.*

Lemma 10 [$\star$]**.** *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be a* Yes *instance of* Group FES*(ℓ) and let S be a minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator in H_I. If S is not regular, then there is an edge $e \in [S]$ which is closed with respect to $R(\hat{Z}_\tau, S)$ and a solution for the given instance containing e^{-1}. Moreover, given the instance and the set S, the arc e^{-1} can be computed in time $\mathcal{O}(\ell(m+n))$.*

This leads us to the following reduction rule.

Reduction Rule 1. *Given an instance $I = (G, \Lambda, k, \hat{Z}, \tau)$ of* Group FES*(ℓ) and a set S which is both irregular and a minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator in H_I, execute the algorithm of Lemma 10 to compute the arc a and return the instance $I' = (G - \{a\}, \Lambda, k-1, \hat{Z}, \tau)$.*

The correctness as well as the fact that it can be applied in time $\mathcal{O}(\ell(m+n))$ follows from Lemma 10. However, observe that in order to apply this rule, we *need* the set S. As a result, we cannot apply this rule exhaustively and we will only apply it selectively during our algorithm. For this, we will also have a slightly modified version of the above reduction rule. We remark that this rule is introduced only in order to maintain a linear dependence on ℓ at certain points in the algorithm.

Reduction Rule 2. *Given an instance $I = (G, \Lambda, k, \hat{Z}, \tau)$ of* GROUP FES*(ℓ) and an arc $a \in A(G)$ such that there is an irregular minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator in H_I that contains 2 edges of $[a]$, return the instance $I' = (G - \{a\}, \Lambda, k-1, \hat{Z}, \tau)$.*

Observe that in the above reduction rule, we have skipped the intermediate step of computing a from the irregular minimum separator. This is because sometimes, we can detect the arc a *faster* than we can compute the irregular minimum separator. We conclude this subsection by arguing that applying these reduction rules to a given instance of GROUP FES(ℓ) does not *increase* the gap. That is, if I' is the instance resulting from I by an application of these rules, then it should not be the case that $\mu(I') > \mu(I)$. In order to prove this (Lemma 12), we need the next lemma.

Lemma 11 [$\star$]**.** *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* GROUP FES*(ℓ). Let $v \in V(G)$, $g \in \Gamma$ and Y be a regular set containing $\hat{Z}_\tau$ such that $\delta(Y)$ is a minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator and let $r = |\delta(Y)|$. Let $e \in \delta(Y)$ be an edge closed with respect to Y and $a = e^{-1}$. Then, the size of a minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator in $H_I - [a]$ is exactly $r - 2$.*

Lemma 12 [$\star$]**.** *Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be an instance of* GROUP FES*(ℓ) and let $I' = (G - \{a\}, \Lambda, k-1, \hat{Z}, \tau)$ be the instance obtained from I by an application of Reduction Rule 1 or Reduction Rule 2 on the arc a. Then, $\mu(I) = \mu(I')$.*

Proof. Recall that $\mu(I) = k - \frac{1}{2}(\lambda_{H_I}(\hat{Z}_\tau, [Z] \setminus \hat{Z}_\tau))$ and $\mu(I') = (k-1) - \frac{1}{2}(\lambda_{H'_I}(\hat{Z}_\tau, [Z] \setminus \hat{Z}_\tau))$. Let $r = \lambda_{H_I}(\hat{Z}_\tau, [Z] \setminus \hat{Z}_\tau)$ and $r' = \lambda_{H'_I}(\hat{Z}_\tau, [Z] \setminus \hat{Z}_\tau)$. Lemma 11 implies that $r' = r - 2$. Hence, $\mu(I') = (k-1) - \frac{1}{2}(r-2) = k - \frac{1}{2}r = \mu(I)$. This completes the proof of the lemma. □

Having stated the reduction rules, we are almost ready to describe the algorithm. Before we do so, we need two subroutines. The first subroutine simply checks whether we can apply Reduction Rule 2 for a given arc. The second one is the main subroutine which either allows us to say NO or apply one of the two reduction rules or compute a pair of 'branchable' arcs or computes an arc which is part of *some* 'farthest' minimum isolating cut separating $\hat{Z}_\tau$ from the rest of $[\hat{Z}]$.

Lemma 13 [$\star$]**.** *There is an algorithm that, given an instance $I = (G, \Lambda, k, \hat{Z}, \tau)$ of* GROUP FES*(ℓ) and an arc $a \in A(G)$, runs in time $\mathcal{O}(k\ell(m+n))$ and correctly* *concludes either that there* exists *a minimum* $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ *separator containing at least 2 edges of* $[a]$ *or that no such separator exists. Furthermore, in the latter case,* $\lambda_{H_I - [a]}(\hat{Z}_\tau, [\hat{Z}] \setminus \hat{Z}_\tau) \geq \lambda_{H_I}(\hat{Z}_\tau, [\hat{Z}] \setminus \hat{Z}_\tau)$*-1.*

Lemma 14 [$\star$](**Main Subroutine**). *There is an algorithm that, given an instance* $I = (G, \Lambda, k, \hat{Z}, \tau)$ *of* Group FES*(ℓ), runs in time* $\mathcal{O}(k^2\ell(m+n))$ *and*

(a) correctly concludes that the given instance is a No *instance or*
(b) returns an irregular minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ *separator of size at most* $2k$ *or an arc* $a \in A(G)$ *such that two edges of* $[a]$ *appear in such a separator or*
(c) returns a pair of edges $e_1, e_2 \in E(H_I)$ *such that there is always a solution intersecting the set* $\{e_1^{-1}, e_2^{-1}\}$ *and there is no irregular minimum* $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ *separator containing at least 2 edges from* $[e_1]$ *or* $[e_2]$ *or*
(d) returns a regular set $Z \supseteq \hat{Z}_\tau$ *such that* $\delta(Z)$ *is a minimum* $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ *separator and an edge* $e^\star = (u_\alpha, v_\beta) \in \delta(Z)$ *where* $u_\alpha \in Z$*, such that there is no* $(Z \cup \{v_\beta\}) - [Z \cup \{v_\beta\}]) \setminus (Z \cup \{v_\beta\})$ *separator of size at most* $|\delta(Z)|$ *and there is no minimum* $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ *separator containing at least 2 edges from* $[e^\star]$*.*

4 Description of the FPT Algorithm for Group FES(ℓ)

Let $I = (G, \Lambda, k, \hat{Z}, \tau)$ be the given instance of Group FES(ℓ). The algorithm we describe is a recursive algorithm with $k = 0$ as the base case. If $k = 0$ then we return Yes if and only if G already has a consistent labelling. Furthermore, if $\lambda_{H_I}(\hat{Z}_\tau, [\hat{Z}] \setminus \hat{Z}_\tau) = 0$ (there is no path in H_I from $\hat{Z}_\tau$ to $[\hat{Z}] \setminus \hat{Z}_\tau$), then we recurse on the instance $(G, \Lambda, k, \emptyset, \tau')$, where τ' is undefined. The correctness of this operation follows from applying Lemma 3 to the connected component of G containing $\hat{Z}$. Finally, if $\hat{Z} = \emptyset$, then we pick an arbitrary vertex v in a component of G which *does not* already have a consistent labeling, pick an arbitrary $g \in \Gamma$ and recurse on the instance $(G, \Lambda, k, \{v\}, \tau')$ where $\tau'(v) = g$. The correctness of this step follows from Observation 2. We now describe the steps executed by the algorithm when $k > 0$ and none of the aforementioned conditions hold. We begin by executing the main subroutine (Lemma 14) on the given instance and describe subsequent steps of the algorithm based on the output of this subroutine.

Case (a): In this case, we simply return No.
Case (b): In this case, we apply Reduction Rule 1 or Reduction Rule 2 as appropriate and recurse on the resulting instance.
Case (c): In this case, we branch on the arcs a_1, a_2 where $a_1 = [e_1^{-1}]$ and $a_2 = [e_2^{-1}]$. That is, we recursively call the algorithm on the tuples $I_1 = (G - a_1, \Lambda, k-1, \hat{Z}, \tau)$ and $I_2 = (G - a_2, \Lambda, k-1, \hat{Z}, \tau)$.
Case (d): Let $Z \supseteq \hat{Z}_\tau$ be the returned regular set and let $e^\star = (u_\alpha, v_\beta) \in \delta(Z)$ be the edge such that $u_\alpha \in Z$, there is no $(Z \cup \{v_\beta\}) - [Z \cup \{v_\beta\}]) \setminus (Z \cup \{v_\beta\}))$ separator of size at most $|\delta(Z)|$ and there is no minimum $\hat{Z}_\tau - [\hat{Z}] \setminus \hat{Z}_\tau$ separator that contains more than one edge from $[e^\star]$. Let $a \in A(G)$ such that $e^\star \in [a]$.

We now branch by either adding a to the solution or by choosing not to pick a in the solution. Formally speaking, we recursively call the algorithm on the tuples $I_1 = (G - a, \Lambda, k - 1, \hat{Z}, \tau)$ and $I_2 = (G, \Lambda, k, \hat{Z} \cup \{v\}, \tau')$ where τ' is the same as τ on $\hat{Z}$ and $\tau'(v) = \beta$.

This completes the description of the algorithm. The correctness of the algorithm follows from the correctness of Lemmas 8 and 14, and the fact that the branching is exhaustive. It remains to analyze the running time. Observe that the time taken at each step of the recursion is dominated by the time required to execute the subroutine of Lemma 14 which runs in time $\mathcal{O}(k^2\ell(m+n))$. Furthermore, along *any* root to leaf path, Reduction Rule 1 applies at most k times. Hence, the running time is bounded by the product of $\mathcal{O}(k^3\ell(m+n))$ and the number of root to leaf paths in the search tree resulting from a run of the algorithm on the input instance $I = (G, \Lambda, k, \hat{Z}, \tau)$.

To complete the proof of Theorem 1, we prove by induction on $\mu(I)$ that the number of leaves in the search tree resulting from a run on input I is bounded by $4^{\mu(I)}$.

5 Concluding Remarks

We have presented an FPT algorithm for Group FES(ℓ) that for finite groups of fixed size has linear dependence on the input-size and matches the best known parameter dependence upto polynomial factors. For this, we had to assume that the multiplication table of the group is explicitly known. A natural question that remains is whether it is possible to obtain a linear time FPT algorithm in the *oracle model* assuming constant query time. Finally, we leave open the question of improving upon the 4^k dependence on the parameter for Group FES(ℓ) even at the cost of superlinear (but still polynomial) dependence on the input-size and group-size.

Acknowledgments. The author acknowledges support from the Austrian Science Fund (FWF), project P26696 X-TRACT.

References

1. Chen, J., Liu, Y., Lu, S.: An improved parameterized algorithm for the minimum node multiway cut problem. Algorithmica **55**(1), 1–13 (2009)
2. Chitnis, R.H., Cygan, M., Hajiaghayi, M., Pilipczuk, M., Pilipczuk, M.: Designing FPT algorithms for cut problems using randomized contractions. In: 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, 20–23 October 2012, pp. 460–469 (2012)
3. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Switzerland (2015)
4. Cygan, M., Pilipczuk, M., Pilipczuk, M.: On group feedback vertex set parameterized by the size of the cutset. Algorithmica **74**(2), 630–642 (2016)

5. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: On multiway cut parameterized above lower bounds. TOCT **5**(1), 3 (2013)
6. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. SIAM J. Comput. **23**(4), 864–894 (1994)
7. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, New York (1999)
8. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin (2006)
9. Geelen, J., Gerards, B.: Excluding a group-labelled graph. J. Comb. Theory Ser. B **99**(1), 247–253 (2009)
10. Guillemot, S.: FPT algorithms for path-transversal and cycle-transversal problems. Discrete Optim. **8**(1), 61–71 (2011)
11. Harary, F.: On the notion of balance of a signed graph. Michigan Math. J. **2**(2), 143–146 (1953)
12. Iwata, Y., Oka, K., Yoshida, Y.: Linear-time FPT algorithms via network flow. In: SODA, pp. 1749–1761 (2014)
13. Iwata, Y., Wahlström, M., Yoshida, Y.: Half-integrality, lp-branching and FPT algorithms (2013). CoRR, abs/1310.2841
14. Kawarabayashi, K., Wollan, P.: Non-zero disjoint cycles in highly connected group labelled graphs. J. Comb. Theory, Ser. B **96**(2), 296–301 (2006)
15. Lokshtanov, D., Narayanaswamy, N.S., Raman, V., Ramanujan, M.S., Saurabh, S.: Faster parameterized algorithms using linear programming. ACM Trans. Algorithms **11**(2), 15:1–15:31 (2014)
16. Marx, D.: Parameterized graph separation problems. Theor. Comput. Sci. **351**(3), 394–406 (2006)
17. Nemhauser, G.L., Trotter Jr., L.E.: Properties of vertex packing and independence system polyhedra. Math. Program. **6**, 48–61 (1974)
18. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford Lecture Series in Mathematics and its Applications, vol. 31. Oxford University Press, Oxford (2006)
19. Ramanujan, M.S., Saurabh, S.: Linear time parameterized algorithms via skew-symmetric multicuts. In: SODA, pp. 1739–1748 (2014)
20. Wahlström, M.: Half-integrality, lp-branching and FPT algorithms. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, 5–7 January 2014, pp. 1762–1781 (2014)
21. Wollan, P.: Packing cycles with modularity constraints. Combinatorica **31**(1), 95–126 (2011)