Discrete Optimization 8 (2011) 61–71

Contents lists available at ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

FPT algorithms for path-transversal and cycle-transversal problems[☆]

Sylvain Guillemot

Institut Gaspard Monge, Université Paris-Est, 5 boulevard Descartes, Champs-sur-Marne, 77454 Marne-la-Vallée, France

ARTICLE INFO

Article history:
Received 15 November 2009
Received in revised form 22 May 2010
Accepted 26 May 2010
Available online 23 June 2010

Keywords:
Cut problems
Feedback set problems
Linear programming
Fixed-parameter algorithms

ABSTRACT

We study the parameterized complexity of several vertex- and edge-deletion problems on graphs, parameterized by the number p of deletions. The first kind of problems are *separation problems* on undirected graphs, where we aim at separating distinguished vertices in a graph. The second kind of problems are *feedback set problems* on group-labelled graphs, where we aim at breaking nonnull cycles in a graph having its arcs labelled by elements of a group. We obtain new FPT algorithms for these different problems, relying on a generic $O^*(4^p)$ algorithm for breaking paths of a *homogeneous* path system.

1. Introduction

Separation and feedback set problems have attracted a lot of interest, first in the area of approximation algorithms, then in parameterized complexity. When parameterized by the number p of deletions allowed, many FPT algorithms were obtained (see [1–3] for an introduction to parameterized complexity and FPT algorithms). Examples of separation problems are the Multiway Cut problem and the Multicut problems, in their vertex- and edge-versions. The first article to address the parameterized complexity of these problems was [4], which obtained an $O^*(4^{p^3})$ algorithm for Multiway Cut, and an $O^*(2^{2kp}p^p4^{p^3})$ algorithm for Multicut with k terminals. Faster algorithms were obtained by [5,6], with [5] presenting the first $O^*(c^p)$ algorithm for Multiway Cut with $c = 4$. Regarding feedback set problems, the most studied is certainly Feedback Vertex Set in undirected graphs, which after a series of improvement was shown to be solvable in $O^*(5^p)$ time [7]. Another example is the Odd Cycle Transversal problem: [8] introduced the method of *iterative compression* to obtain an $O^*(3^p)$ algorithm for the problem. In the case of directed graphs, [9] obtained an FPT algorithm for the Directed Feedback Vertex Set problem, settling a long-standing conjecture.

In this article, we obtain new parameterized algorithms for the aforementioned separation problems, as well as for new feedback set problems involving group-labelled graphs. Our results are as follows. For the Multiway Cut, we obtain a new algorithm with an $O^*(4^p)$ running time similar to [5]. For the Multicut problems, we obtain simple algorithms with running time $O^*((8k)^p)$, improving upon the results of Marx. We then consider problems on group-labelled graphs. If Γ is a finite group, a *Γ-labelled graph* is a digraph whose arcs are labelled by elements of Γ, and satisfying an additional symmetry property. These objects appear under different names in the literature; see e.g. [10,11] for recent results. We consider the Group Feedback Set problems, where the input is a Γ-labelled graph, and where the goal is to break all nonnull cycles by p deletions. We consider the problems in their vertex- and edge-versions. We show that both versions are solvable in $O^*((4|\Gamma| + 1)^p)$ time, and that the edge-version is also solvable in $O^*((8p + 1)^p)$ time, independently of Γ. These results

[☆] An extended abstract of this article was published in Proceedings of the 3rd International Workshop on Parameterized and Exact Computation, IWPEC 2008, Lecture Notes in Computer Science, vol. 5018, Springer-Verlag, 2008, pp. 129–140.

E-mail address: sylvain.guillemot@univ-mlv.fr.

1572-5286/$ – see front matter
doi:10.1016/j.disopt.2010.05.003

generalize the FPT algorithms for the Odd Cycle Transversal problem, which is a particular case of the Group Feedback Set problem with $\Gamma = \mathbb{Z}_2$.

These different algorithms use as subroutine a generic $O^*(4^p)$ algorithm for breaking a set of paths in a graph, provided that the set has a special property called *homogeneity*. Formally, we call *path system* a tuple σ consisting of an undirected graph $G = (V, E)$, a set $T \subseteq V$ of terminals, a set $F \subseteq V$ of forbidden vertices, and a set $\mathcal{P}$ of paths between terminals. The generic Path Transversal problem aims at breaking each path $P \in \mathcal{P}$ of a path system σ by removing nonforbidden vertices. We define *homogeneous* path systems, and we address the approximability and the fixed-parameter tractability of the Path Transversal problem on homogeneous instances. Namely, we show that the problem has a half-integrality property generalizing [12], which yields a 2-approximation. We then devise a bounded-search algorithm which solves the problem in $O^*(4^p)$ time, by relying on half-integral solutions in order to guide the construction of a search tree. To our knowledge, this is the first example of an FPT algorithm which uses fractional LPs.

This article is organized as follows. Section 2 is devoted to our generic algorithm for homogeneous path systems. Section 3 contains results for the Multiway Cut and Multicut problems. Section 4 contains results for the Group Feedback Set problems. Finally, in Section 5 we formulate some open questions and possible generalizations of the results.

2. Homogeneous path systems

2.1. Preliminaries

Let $G = (V, E)$ be an undirected graph. A *path* in G is a sequence of vertices $P = x_1, \ldots, x_m$ s.t. $x_ix_{i+1} \in E$ for each $1 \leq i < m$; we say that P is a path joining x_1 to x_m. A *cycle* in G is a path $x_1x_2, \ldots, x_m$ with x_1, x_m equal; we say that C is a cycle at x_1. Consider a path $P = x_1, \ldots, x_m$. The vertices $x_1, \ldots, x_{m-1}$ are the *initial vertices* of P. The *inverse* of P is the path $P^{-1} = x_m, \ldots, x_1$. Given a weight function w on V, the *length* of P is $w(x_1) + \cdots + w(x_m)$, and the *initial length* of P is $w(x_1) + \cdots + w(x_{m-1})$. We say that P is *simple* iff the vertices x_i are distinct. Henceforth, paths and cycles are not necessarily simple, unless explicitly stated.

A *path system* is a tuple $\sigma = (G, T, F, \mathcal{P})$ which consists of: (i) an undirected graph $G = (V, E)$, (ii) a set $T \subseteq V$ of *terminals*, (iii) a set $F \subseteq V$ of *forbidden vertices*, (iv) a set $\mathcal{P}$ of paths in G joining elements of T. A *transversal* of σ (or a *solution for* σ) is a set of vertices disjoint from F and which meets each path of $\mathcal{P}$. The generic problem Path Transversal takes an instance $I = (\sigma, p)$ consisting of a path system σ, an integer p, and seeks a transversal of σ of size at most p.[1]

In this section, we consider the restriction of the Path Transversal problem to *homogeneous* instances. This notion is defined as follows:

Definition 1. The path system $\sigma = (G, T, F, \mathcal{P})$ is *homogeneous* iff the two following conditions hold:

1. for each path $P \in \mathcal{P}$, there exists a simple path $P' \in \mathcal{P}$ included in P;
2. for each path $P \in \mathcal{P}$, if $P = P_1xP_2$ then: for each path P' joining T to x, one of $P_1P'^{-1}, P'P_2$ is in $\mathcal{P}$.

This seemingly abstract notion arises from a close examination of the proof of the half-integrality of Multiway Cut in [12]. While enjoyed by the Multiway Cut problem itself, this property also applies to other problems, as exemplified in Section 4. In this section, we address the solvability of the Path Transversal problem for homogeneous instances. First, we show in Section 2.2 that it admits a half-integral LP formulation, by adapting the proof from [12]. Second, we show in Sections 2.3 and 2.4 that the property gives rise to an $O^*(4^p)$ time algorithm for the problem.

2.2. LP formulation and half-integrality

In this subsection, we describe the LP formulation of the problem, and we demonstrate its half-integrality. The Path Transversal problem can be formulated by the following LP, which will be denoted by F_σ.

$$\text{minimize } \sum_{v \in V} d_v$$

$$\text{such that } \forall P \in \mathcal{P}, \sum_{v \in P} d_v \geq 1$$

$$\forall v \in V, d_v \geq 0, \forall v \in F, d_v = 0.$$

It is easy to see that the integral solutions of F_σ coincide with the solutions of the Path Transversal problem. The dual LP of F_σ is the following program, which will be denoted by F'_σ.

[1] Note that in the description of the instance I, the set $\mathcal{P}$ will not be described in extension, since it may be of exponential size. Instead, we will assume that $\mathcal{P}$ is described by a particular oracle; see Section 2.2.

$$\text{maximize} \sum_{p \in \mathcal{P}} f_p$$

$$\text{such that } \forall v \in V \setminus F, \sum_{p \in \mathcal{P}: v \in p} f_v \leq 1$$

$$\forall p \in \mathcal{P}, f_p \geq 0.$$

For our algorithmic purposes, we will only need to solve F_σ. As is well known, the optimal solution to this fractional LP can be found in polynomial time, provided that we have a polynomial-time *separation oracle*. Recall that a separation oracle is an algorithm which, given values $(d_v)_{v \in V}$, either concludes that it is an optimal solution or finds a violated constraint. In our case, the only nontrivial constraint is the first one; so we will be interested in finding a shortest path $P \in \mathcal{P}$ under the weighting d.

We now demonstrate a half-integrality property of F_σ. While our proof closely follows the proof of [12], we include it for completeness. Given a solution d of F_σ, let G_d denote the vertex-weighted graph obtained from G by weighting each $v \in V$ by d_v. Let d be an optimal solution of F_σ, and let f be the corresponding optimal solution of F'_σ. Let M be the set of $v \in V$ s.t. $d_v > 0$ and v is reachable from T by a path of initial length 0 in G_d. Let $V_1 = \{v \in M : d_v = 1\}$, $V_{1/2} = \{v \in M : 0 < d_v < 1\}$, and $V_0 = V \setminus (V_1 \cup V_{1/2})$.

Lemma 1. *Let $P \in \mathcal{P}$ s.t. $f_P > 0$. Then $P \cap M$ consists of: either an element of V_1, or two elements of $V_{1/2}$.*

Proof. Since $f_P > 0$, dual complementary slackness implies that P has length one in G_d. Let u be the first vertex of P s.t. $d_u > 0$, and let v be the last vertex of P s.t. $d_v > 0$. Then $u, v \in M$. If $u = v$, since P has length one then $d_u = 1$, and thus $P \cap M$ consists of a single element of V_1. Suppose now that u, v are distinct, then they belong to $V_{1/2}$.

Aiming for contradiction, suppose that there exists a third vertex $w \in P \cap M$. We can then write $P = P_1 w P_2$, where P_1 contains u and P_2 contains v. Since $w \in M$, there exists a path P' joining T to w, such that P' has initial length 0 in G_d. By Point 2 of Definition 1, one of $P_1 P'^{-1}$, $P' P_2$ is in $\mathcal{P}$. But these paths have length less than 1 in G_d, contradicting the assumption that d is a solution of F_σ. □

Let s be the solution of F_σ which assigns the value r to a vertex of V_r, for every $r \in \{0, \frac{1}{2}, 1\}$.

Lemma 2. *s is an optimal solution of F_σ.*

Proof. Let us first show that s is a solution of F_σ. Consider $P \in \mathcal{P}$. Since P has length ≥ 1 in G_d, P contains at least one vertex of M. Then either P contains a vertex of V_1, or P contains two vertices $V_{1/2}$. In both cases, we obtain that $\sum_{v \in P} s_v \geq 1$.

We now show that s is optimal by proving that f has the same cost as s. We define two subsets $\mathcal{P}_1$, $\mathcal{P}_2$ of $\mathcal{P}$ as follows: (a) $\mathcal{P}_1$ is the set of paths $P \in \mathcal{P}$ s.t. $f_P > 0$ and $P \cap M$ consists of an element of V_1, (b) P_2 is the set of paths $P \in \mathcal{P}$ s.t. $f_P > 0$ and $P \cap M$ consists of two elements of $V_{1/2}$. By primal complementary slackness and by the optimality of d, each vertex of M is saturated. It follows that:

$$\sum_{P \in \mathcal{P}} f_P = \sum_{P \in \mathcal{P}_1} f_P + \sum_{P \in \mathcal{P}_2} f_P = |V_1| + \frac{1}{2}|V_{1/2}|$$

where the last equality follows by summing the contributions of the vertices of M, and using Lemma 1. We conclude by observing that this is exactly the cost of s. □

2.3. A technical result

This subsection is devoted to the proof of Proposition 1, which states the existence of optimal solutions with specific properties. Let U be the set of elements of V_0 reachable from T by a path of length 0 in G_s.

Proposition 1. *There is an optimal solution for σ disjoint from U.*

Let S be an optimal solution for σ. We define the set of bad vertices $B = S \cap U$. Our goal is to construct a solution S' from S by discarding the bad vertices and replacing them by some vertices outside of U.

Let $u \in V$. Say that u is *accessible* iff it is reachable from T by a path P such that (i) P goes through an element of B, (ii) P has initial length 0 in G_s. Say that u is *uniformly accessible* if for each path P joining T to u, if P has initial length 0 in G_s then P goes through an element of B. We define V'_1 as the set of accessible elements of $V_1 - S$, and we define $V'_{1/2}$ as the set of uniformly accessible elements of $V_{1/2} - S$. We set $V' = V'_1 \cup V'_{1/2}$ and $S' = S - B + V'$.

We will show that S' is an optimal solution for σ disjoint from U. We need the two following lemmas.

Lemma 3. *Let $P \in \mathcal{P}$ s.t. P does not intersect $S - B$. Then P intersects V'.*

Proof. Let $n(P)$ denote the number of elements of $P \cap M$. We reason by induction on $n(P)$.

Case 1: $n(P) = 1$. The set $P \cap M$ then consists of an element $u \in V_1 - S$. But $P = P_1 u P_2$ is covered by an element $v \in B$, and since $v \neq u$ we can assume that $v \in P_1$. Now, the path $P_1 u$ joins T to u, goes through $v \in B$ and has initial length 0 in G_s. This implies that u is accessible, and we conclude that $u \in V'_1$.

Case 2: $n(P) \geq 2$. Let u_1, u_2 be the first and last elements of $P \cap M$, respectively. We have $P = P_1 u_1 P_2 u_3 P_3$. We consider two subcases, according to whether P_2 intersects B or not.

Case 2.1: $P_2 \cap B \neq \emptyset$. Consider an element $v \in P_2 \cap B$, and let u'_1 be the last element of $P \cap M$ preceding v, and u'_2 be the first element of $P \cap M$ following v. We then have $P = P'_1 u'_1 P'_2 u'_2 P'_3$, with $P'_2 = P''_1 v P''_2$. Since $v \in B$, there exists a path P' joining T to v and having length 0 in G_s. By Point 2 of Definition 1, one of $P'_1 u'_1 P''_1 P'^{-1}$, $P' P''_2 u'_2 P'_3$ is a bad path. Let P'' be this path, then P'' does not intersect $S - B$ (since neither P nor P' do), and $n(P'') < n(P)$. We apply the induction hypothesis to obtain that P'' intersects V'. Since P' is disjoint from V', it follows that P intersects V'.

Case 2.2: $P_2 \cap B = \emptyset$. If $u_1 \in V'$ or $u_2 \in V'$, we are done. Suppose now that $u_1, u_2 \notin V'$. Then one of P_1, P_3 must intersect B, and by symmetry we assume that $P_1 \cap B \neq \emptyset$. It follows that u_1 is accessible; since $u_1 \notin V'$, we cannot have $u_1 \in V_1$ (since it would imply $u_1 \in V'_1$) and we thus have $u_1 \in V_{1/2} \setminus V'_{1/2}$. Then u_2 is not uniformly accessible, and there exists a path P'_1 joining T to u_1, having initial length 0 in G_s, and avoiding B. By Point 2 of Definition 1, one of $P_1 P'^{-1}_1$, $P'_1 P_2 u_2 P_3$ is a bad path. Since $u_1 \notin V_1$, the former has length <1 in G_s and cannot be a bad path. It follows that the path $P'_1 P_2 u_2 P_3$ is bad, and thus it has to be covered by B; since P'_1, P_2 are not we obtain that $P_3 \cap B \neq \emptyset$. By the same reasoning as for u_1, we find a path P'_2 joining T to u_2, avoiding B, such that $P'_1 P_2 P'^{-1}_2$ is a bad path. But this latter path is not covered by B, impossible. □

Let $\epsilon > 0$. Define s' from s by assigning $s_v - \epsilon$ to $v \in V'$, $s_v + \epsilon$ to $v \in B$, s_v to any other vertex v.

Lemma 4. *For ϵ small enough, s' is a solution of F_σ.*

Proof. Consider $P \in \mathcal{P}$, we show that $\sum_{v \in P} s'_v \geq 1$. By Point 1 of Definition 1, it suffices to show it for P simple, which we assume in the following. If $\sum_{v \in P} s_v > 1$, then the desired inequality holds for ϵ small enough. Suppose now that $\sum_{v \in P} s_v = 1$. Since s is half-integral, $P \cap M$ contains either one vertex of V_1 or two vertices of $V_{1/2}$.

If $P \cap M$ consists of exactly one vertex $v \in V_1$, then so does P', and thus $P' = P'_1 v P'_2$ with P'_1, P'_2 disjoint from M. We have the following subcases. If $v \notin V'_1$, then P is disjoint from V' and thus $\delta \geq 0$. If $v \in V'_1$, since $v \notin S$ it follows that P is covered by an element $u \in S$ appearing in P'_1 or P'_2. Now, $u \notin M$ implies that u is in V_0 and thus in U. We obtain that $\delta \geq \epsilon - \epsilon \geq 0$.

If $P \cap M$ consists of exactly two vertices $u, v \in V_{1/2}$, then so does P', hence P' has the form $P'_1 u P'_2 v P'_3$. We have the following subcases. If $u, v \notin V'_{1/2}$, then P is disjoint from V' and thus $\delta \geq 0$. If $u \in V'_{1/2}$, $v \notin V'_{1/2}$: since u is uniformly accessible, P'_1 contains a vertex from B, hence $\delta \geq \epsilon - \epsilon \geq 0$. If $u \notin V'_{1/2}$, $v \in V'_{1/2}$: similar to the previous case. If $u, v \in V'_{1/2}$: since u, v are both uniformly accessible, both P'_1, P'_3 contain a vertex from B, and these vertices are distinct since P' is simple, which implies that $\delta \geq 2\epsilon - 2\epsilon \geq 0$. □

We are now ready to show that S' is an optimal solution for σ disjoint from U. First, S' is a solution for σ by Lemma 3. Second, S' is disjoint from U, since V' is disjoint from V_0 and thus from U. Finally, we show that $|S'| \leq |S|$ by proving that $|V'| \leq |B|$. Suppose by contradiction that $|V'| > |B|$, then by Lemma 4 we obtain s' solution of F_σ s.t. $|s'| = |s| + \epsilon(|B| - |V'|) < |s|$. This contradicts the optimality of s.

2.4. The main result

In this subsection, we describe the algorithm for Path Transversal. We will justify its correctness and running time in Theorem 1.

Let $\sigma = (G, T, F, \mathcal{P})$ be a homogeneous path system. We denote the cost of an optimal fractional solution of F_σ by opt^*_σ, and we denote the cost of an optimal solution for σ by opt_σ. By convention, these values are equal to ∞ when there is no solution. A *frontier vertex* is a vertex u s.t. (i) $u \notin F$, (ii) u is reachable from T by a path whose initial vertices are in F.

The description of the algorithm follows:

The following lemma ensures the correctness of the branching strategy in Line 15.

Lemma 5. *(σ, p) is a positive instance iff one of (σ', p), $(\sigma'', p - 1)$ are positive instances.*

Proof. Let S be a solution for σ of size $\leq p$. If $u \notin S$, then S is a solution for σ'. If $u \in S$, then $S \setminus \{u\}$ is a solution for σ''.

Conversely, if S is a solution for σ' of size $\leq p$, then it is a solution for σ. If S is a solution for σ'' of size $\leq p - 1$, then $S \cup \{u\}$ is a solution for σ. □

The crucial fact is that when $opt^*_\sigma = opt^*_{\sigma'}$, no branching is needed (cf. Lines 11–12). This relies on Proposition 1.

Lemma 6. *If $opt^*_\sigma = opt^*_{\sigma'}$, then $opt_\sigma = opt_{\sigma'}$ (and thus the instances (σ, p) and (σ', p) are equivalent).*

Proof. Since any solution for σ' is also a solution for σ, we have $opt_{\sigma'} \geq opt_\sigma$. We now show that $opt_{\sigma'} \leq opt_\sigma$. Let s be an half-integral optimal solution of $F_{\sigma'}$, and let $V_0, V_{1/2}, V_1$ be defined accordingly. Observe that s is also a solution of F_σ. Moreover, it is an *optimal* solution of F_σ since $opt^*_\sigma = opt^*_{\sigma'}$.

Let U be the set of elements of V_0 reachable from T by a path of length 0 in G_s. By Proposition 1, there exists S optimal solution for σ disjoint from U. Since u is a frontier vertex, we then have $u \in U$, hence $u \notin S$. It follows that S is a solution for σ', and we conclude that $opt_{\sigma'} \leq |S| = opt_\sigma$. □

```
SolvePT(σ, p)
1: suppose that σ = (G, T, F, 𝒫)
2: x ← opt*_σ
3: if x ≤ p/2 then
4:     return true
5: else if x > p then
6:     return false
7: end if
8: choose u a frontier vertex
9: let σ′ = (G, T, F ∪ {u}, 𝒫)
10: x′ ← opt*_σ′
11: if x′ = x then
12:     return SolvePT(σ′, p)
13: else if x′ > x then
14:     let σ″ = (G\u, T, F, 𝒫)
15:     return (SolvePT(σ′, p) or SolvePT(σ″, p − 1))
16: end if
```

With these two lemmas, we are ready to prove the correctness and running time of the algorithm.

Theorem 1. *Suppose that σ is homogeneous and admits a polynomial-time separation oracle. Then* SolvePT (σ, p) *solves, in $O^*(4^p)$ time, the* Path Transversal *problem on the instance (σ, p).*

Proof. *Correctness.* The case when $x \leq \frac{p}{2}$ or $x > p$ is handled correctly by the algorithm: since $opt^*_\sigma \leq opt_\sigma \leq 2opt^*_\sigma$, we have $opt_\sigma \leq p$ in the first case, and $opt_\sigma > p$ in the second case. Suppose now that $\frac{p}{2} < x \leq p$. Observe that there exists a frontier vertex: otherwise, each path in $\mathcal{P}$ would only contain nodes in F, and we would have $x = \infty$. In Lines 11–12, the case $x' = x$ is handled correctly by Lemma 6. In Lines 13–16, the case $x' > x$ is correct by Lemma 5.

Running time. We view the execution of the algorithm as a search tree, where only recursive calls in Line 15 correspond to branches in the search tree. A node of the search tree is labelled by an instance (σ, p). If we let $P(n)$ denote the running time of the operations of Lines 1–10, then the processing time of a node is bounded by $nP(n)$. This follows from the fact that at each recursive call in Line 12, $|F|$ increases by one and is upper bounded by n.

Let T denote the search tree, and let u be a node of T. Suppose that u is labelled by (σ, p), let $p(u) = p$ and $k(u) = 2p + 1 - 2opt^*_\sigma$. Observe that if u is an internal node of T, then $\frac{p}{2} < opt^*_\sigma \leq p$, and thus $0 < k(u) \leq p$. Let $S(u)$ denote the number of leaves in the subtree of T rooted at u. Given p, k, let $S(p, k)$ denote the maximum value of $S(u)$ for u node of T s.t. $p(u) = p$, $k(u) \leq k$, or 0 if no such node exists. We claim that:

$$\begin{cases} S(p,k) = 1 & \text{if } p = 0 \text{ or } k = 0 \\ S(p,k) \leq S(p,k-1) + S(p-1,k) & \text{otherwise.} \end{cases} \tag{1}$$

Let u be a node labelled by (σ, p). If u is a leaf, then $S(u) = 1$. Observe that this holds in particular if $p = 0$ or $k = 0$: this is clear if $p = 0$, this results from the fact that $opt^*_\sigma > p$ if $k = 0$. Suppose now that u is an internal node with two children u', u'', with u' labelled by (σ', p) and u'' labelled by $(\sigma'', p - 1)$. We then have $k, p > 0$, and we need to bound $S(u')$ and $S(u'')$.

We first bound $S(u')$. Since $opt^*_{\sigma'} > opt^*_\sigma$ and since $opt^*_{\sigma'}$ is half-integral, we have $k(u') \leq k(u) - 1 \leq k - 1$, hence $S(u') \leq S(p, k - 1)$. We now bound $S(u'')$. Observe that $opt^*_{\sigma''} \geq opt^*_\sigma - 1$: indeed, given s solution for $F_{\sigma''}$ of cost c, we can extend it to V by setting $s_u = 1$, obtaining a solution for F_σ of cost $c + 1$. It follows that $k(u'') = 2(p - 1) + 1 - 2opt^*_{\sigma''} \leq 2(p - 1) + 1 - 2(opt^*_\sigma - 1) = 2p + 1 - 2opt^*_\sigma \leq k$, and we obtain that $S(u'') \leq T(p - 1, k)$.

We thus conclude that $S(u) = S(u') + S(u'') \leq S(p, k - 1) + S(p - 1, k)$, which completes the proof that S satisfies the relations (1). A straightforward induction then shows that $S(p, k) \leq 2^{p+k}$. Since $k(u) \leq p$ for an internal node u of T, it follows that the number of leaves of T is bounded by $S(p, p) \leq 2^{2p}$. Since each node of T is processed in polynomial time, we obtain the claimed $O^*(4^p)$ running time. □

3. New algorithms for separation problems

3.1. *The* Multiway Cut *problems*

Let $G = (V, E)$ be a graph and let $T \subseteq V$ be a set of terminals. Given two partitions $\mathcal{S}, \mathcal{S}'$ of T, we write $\mathcal{S} \sqsubseteq \mathcal{S}'$ to mean that $\mathcal{S}$ refines $\mathcal{S}'$. Given $x, y \in V$, we write $x \equiv_\mathcal{S} y$ iff x, y belong to the same class of $\mathcal{S}$. We denote by $C_T(G)$ the partition $\mathcal{S}$ of T whose classes are the sets $C \cap T$ for C connected component of G. If F is a forest, we denote by $F|T$ the forest obtained from F by repeatedly performing the following operations: (i) remove the leaves not belonging to T, (ii) contract the nodes of degree 2 not belonging to T.

We consider the Generalized Vertex Multiway Cut (GVMC) problem: given a graph $G = (V, E)$, a set $T \subseteq V$ of terminals, a set $F \subseteq V$ of forbidden vertices, a partition $\mathcal{S}$ of T, an integer p, can we find a set $S \subseteq V \setminus F$ of size $\leq p$ s.t. $C_T(G \setminus S) \sqsubseteq \mathcal{S}$. In other terms, we want to disconnect pairs of terminals lying in different classes of $\mathcal{S}$. We also consider the edge-version of the problem called Generalized Multiway Cut (GMC). While this was already shown in [5], we obtain another proof of the following result:

Theorem 2. Generalized Vertex Multiway Cut *and* Generalized Multiway Cut *are solvable in* $O^*(4^p)$ *time.*

Proof. We formulate the GVMC problem as a path transversal problem for a homogeneous path system, and apply Theorem 1. We consider the path system $\sigma = (G, T, F, \mathcal{P})$, where $\mathcal{P}$ consists of the paths joining pairs of terminals belonging to different classes of $\mathcal{S}$.[2] We verify that σ is homogeneous and has a separation oracle:

(a) Point 1 of Definition 1: if $P \in \mathcal{P}$ joins $u, v \in T$ with $u \not\equiv_{\mathcal{S}} v$, then we find a simple path $P' \subseteq P$ joining u, v. Clearly, P' is also in $\mathcal{P}$.
(b) Point 2 of Definition 1: suppose that $P \in \mathcal{P}$ joins $u, v \in T$ with $u \not\equiv_{\mathcal{S}} v$. Suppose that $P = P_1xP_2$ and consider P' joining $w \in T$ to x. Since we cannot have both $w \equiv_{\mathcal{S}} u$ and $w \equiv_{\mathcal{S}} v$, it follows that one of $P_1P'^{-1}, P'P_2$ is in $\mathcal{P}$.
(c) σ has a polynomial-time separation oracle: indeed, if we are given $(d_v)_{v \in V}$, we compute the all-pairs shortest paths in G_d. A shortest bad path is then obtained by examining the computed distances for each pairs of vertices $u, v \in T$ belonging to different classes of $\mathcal{S}$.

An algorithm for GMC is obtained by a simple reduction to GVMC. Given $I = (G, T, F, \mathcal{S}, p)$ instance of GMC, we construct $I' = (G', T, F', \mathcal{S}, p)$ instance of GVMC as follows. For each edge e of G we introduce a new vertex x_e. We subdivide each edge $e = uv$ in two edges ux_e, vx_e. We set $F' = V \cup \{x_e : e \in F\}$. It is easy to see that the function which maps a set $S \subseteq E$ to $S' = \{x_e : e \in S\}$ induces a bijection between the solutions of I' and the solutions of I. $\square$

3.2. *The* Multicut *problem*

We now consider the Multicut problem: given a graph $G = (V, E)$, a set $T \subseteq V$ of terminals, a set $P \subseteq [T]^2$ of pairs of terminals, a set $F \subseteq E$ of forbidden edges, an integer p, can we disconnect each pair of vertices in P by removing at most p edges of $E \setminus F$? We obtain an FPT algorithm for the problem when the number of terminals $k := |T|$ is bounded.

Theorem 3. Multicut *can be solved in* $O^*((8k)^p)$ *time.*

Proof. Let $I = (G, T, P, F, p)$ be an instance of Multicut. Say that a partition $\mathcal{S}$ of T is *realizable* iff (a) $\mathcal{S} \sqsubseteq C_T(G)$, and (b) $I' = (G, T, F, \mathcal{S}, p)$ is a positive instance of GMC. Say that a partition $\mathcal{S}$ of T is *admissible* iff it separates each pair in P. Observe that solving I amounts to find a partition $\mathcal{S}$ of T which is admissible and realizable.

We will describe an algorithm that enumerates a set $\mathcal{G}$ of *good* partitions of T s.t. (i) $\mathcal{G}$ contains the realizable partitions; (ii) $\mathcal{G}$ has size $\leq (2k)^p$. This algorithm will allow us to solve Multicut as follows: for each partition $\mathcal{S} \in \mathcal{G}$, we test in polynomial time if $\mathcal{S}$ is admissible, we test in $O^*(4^p)$ time if the partition is realizable (by Theorem 2), and we accept the instance I as soon as we have found $\mathcal{S}$ which passes the two tests. Therefore, the total running time of the algorithm is $O^*((8k)^p)$ as claimed.

The set $\mathcal{G}$ is the set of partitions returned by the computation paths of the following nondeterministic algorithm:

FindGoodPartition(G, T, p)
1: let F be a spanning forest of G, let $F' = F|T$
2: choose a set E of at most p edges of F'
3: let $\mathcal{S} = C_T(F')$, let $\mathcal{S}' = C_T(F' \backslash E)$
4: choose a partition $\mathcal{S}''$ s.t. $\mathcal{S}' \sqsubseteq \mathcal{S}'' \sqsubseteq \mathcal{S}$
5: return $\mathcal{S}''$

We first show that $\mathcal{G}$ has property (i). Let $\mathcal{S}_0$ be a realizable partition, we show that it is returned by some computation path of the algorithm. Observe first that $\mathcal{S} = C_T(G)$, hence $\mathcal{S}_0 \sqsubseteq \mathcal{S}$ by Point (a) of the definition of a realizable partition. Now, suppose that $\mathcal{S}_0$ verifies Point (b) by removing a set S of at most p edges of G. To S there corresponds a set S' of at most p edges of F' s.t. $C_T(F' \setminus S') = C_T(F \setminus S)$. Note that $C_T(F \setminus S) \sqsubseteq C_T(G \setminus S) \sqsubseteq \mathcal{S}_0$ by Point (b). Thus, it follows that $C_T(F' \setminus S') \sqsubseteq \mathcal{S}_0$. Hence, by considering the execution of the algorithm which chooses $E = S'$, we have $\mathcal{S}' \sqsubseteq \mathcal{S}_0 \sqsubseteq \mathcal{S}$. We conclude that $\mathcal{S}_0$ is returned by some computation path having chosen $E = S'$.

We now show that $\mathcal{G}$ has property (ii). We first observe that: if $\mathcal{S}$ has m classes, $\mathcal{S}'$ has m' classes, and $\mathcal{S}' \sqsubseteq \mathcal{S}$, then the number of partitions $\mathcal{S}''$ s.t. $\mathcal{S}' \sqsubseteq \mathcal{S}'' \sqsubseteq \mathcal{S}$ is $\leq B(m' - m) \leq (m' - m)!$ (where B is the Bell number). Now, it must hold that $m' \leq m + p$, since removing an edge can increase the number of connected components of F' by at most one. Therefore, there are at most $p!$ possible choices in Line 4. Since F has at most $2k$ edges, there are at most $\binom{2k}{p} \leq \frac{(2k)^p}{p!}$ choices in Line 2. Overall, the total number of computation paths of the algorithm is at most $(2k)^p$. $\square$

[2] $\mathcal{P}$ is the set of $\mathcal{S}$-paths in the sense of Mader.

3.3. *The* Vertex Multicut *problem*

We finally consider the vertex-version of the Multicut problem, called Vertex Multicut: given a graph $G = (V, E)$, a set $T \subseteq V$ of terminals, a set $P \subseteq [T]^2$ of pairs of terminals, a set $F \subseteq V$ of forbidden vertices, can we disconnect each pair of vertices in P by removing at most p vertices in $V \setminus F$? We obtain an FPT algorithm when $k := |T|$ is bounded, using similar ideas to Theorem 3.

Theorem 4. Vertex Multicut *can be solved in $O^*((8k)^p)$ time.*

Proof. Let $I = (G, T, P, F, p)$ be an instance of Vertex Multicut. We now say that a partition $\mathcal{S}$ of T is realizable iff (a) $\mathcal{S} \sqsubseteq C_T(G)$ and (b) $I' = (G, T, F, \mathcal{S}, p)$ is a positive instance of GVMC. As before, we describe an algorithm that enumerates a set $\mathcal{G}$ of good partitions of T s.t. (i) $\mathcal{G}$ contains the realizable partitions, (ii) $\mathcal{G}$ has size $\leq (p+1)(2k)^p$. This gives rise to an $O^*((8k)^p)$ algorithm for Vertex Multicut, by the same argument as above.

The set $\mathcal{G}$ is obtained as the results of the computation paths of the following nondeterministic algorithm:

```
FindGoodPartition2(G, T, p)
1: let F be a spanning forest of G, let F' = F|T
2: let N be the set of nodes of F'
3: choose i ∈ {1, 2}
4: if i = 1 or p = 0 then
5:    return FindGoodPartition(G, T, p)
6: else
7:    choose u ∈ N
8:    return FindGoodPartition2(G\u, T, p − 1)
9: end if
```

We first show that $\mathcal{G}$ has property (i). We prove by induction on p that: if $\mathcal{S}_0$ is realizable by removing p vertices, then it is returned by some computation path of FindGoodPartition2 (G, T, p). Suppose that $\mathcal{S}_0$ is obtained by removing a set S of at most p vertices of G. Let F, F' as chosen by FindGoodPartition2 (G, T, p).

If $S \cap N \neq \emptyset$, then consider an element u in the intersection. For the graph $G \setminus u$, $\mathcal{S}_0$ is realizable by removing $p-1$ vertices, hence by induction hypothesis it is returned by some computation path of FindGoodPartition2 $(G \setminus u, T, p-1)$. This path is completed in a computation path of FindGoodPartition2 (G, T, p) which chooses $i = 2$ in Line 3 and u in Line 7.

Suppose now that $S \cap N = \emptyset$. By definition of F', to each edge $e = uv$ of F' there corresponds a path P_e in F whose endpoints are u, v and whose internal nodes are in $V \setminus N$. Let S' be the set of edges e of F' s.t. P_e intersects S. Since the P_e are internally disjoint, it follows that $|S'| \leq |S| \leq p$. Let $\mathcal{S} = C_T(F')$ and $\mathcal{S}' = C_T(F' \setminus S')$. As in Theorem 3, we have $\mathcal{S}_0 \sqsubseteq \mathcal{S}$ by Point (a) of the definition of a realizable partition. Besides, by definition of S', from a path in $F' \setminus S'$ joining $u, v \in T$, we obtain a path in $F \setminus S$ joining u, v, which implies that $\mathcal{S}' \sqsubseteq \mathcal{S}_0$ by Point (b). It follows that $\mathcal{S}_0$ is returned by an execution of FindGoodPartition (G, T, p) which chooses $E = S'$. This execution is completed in an execution of FindGoodPartition2 (G, T, p) which chooses $i = 1$ in Line 3.

We now show that $\mathcal{G}$ has property (ii). Let $T_2(p)$ denote the maximum number of computation paths of the algorithm FindGoodPartition2 (G, T, p). Since F' has at most $2k$ vertices, there are at most $2k$ possible choices in Line 7, hence T_2 satisfies the following relation:

$$\begin{cases} T_2(0) = 1 \\ T_2(p) \leq T_1(p) + (2k)T_2(p-1). \end{cases}$$

Recall that $T_1(p) \leq (2k)^p$ from Theorem 3. Therefore, a straightforward induction gives the desired bound $T_2(p) \leq (p+1)(2k)^p$. □

4. Problems on group-labelled graphs

4.1. Preliminaries

Let Γ be a finite group, with unit element 1_Γ. In the following, we will assume that Γ is described by its multiplication table, of size $O(|\Gamma|^2)$. In the running time analysis, the factors $O(|\Gamma|^c)$ will be assumed constant and thus will be omitted.

A *Γ-labelled graph* is a digraph with a labelling of its arc by elements of Γ. Formally, this is a tuple $\mathcal{G} = (V, A, \Lambda)$, where V is a set of vertices, $A \subseteq V^2$ is a set of arcs, and $\Lambda : A \to \Gamma$ is a labelling of the arcs, satisfying the following property: if $(u, v) \in A$, then $(v, u) \in A$ and $\Lambda(v, u) = \Lambda(u, v)^{-1}$.

The *underlying graph* of $\mathcal{G}$ is the undirected graph $G = (V, E)$ where $E = \{uv : (u, v) \in A\}$. By a path (or cycle) in $\mathcal{G}$, we will mean a path (or cycle) in G. Let $P = x_1, \ldots, x_m$ be a path in $\mathcal{G}$, we set $\Lambda(P) = \Lambda(x_1, x_2), \ldots, \Lambda(x_{m-1}, x_m)$ (or $\Lambda(P) = 1_\Gamma$ if $m = 1$). A cycle in $\mathcal{G}$ is *null* if $\Lambda(C) = 1_\Gamma$, *nonnull* otherwise.

We observe that nonnull cycles are well defined even if Γ is nonabelian:

Lemma 7. *If $C = x_1, \ldots, x_m x_1$ is a nonnull cycle at x_1, then $C' = x_2, \ldots, x_1 x_2$ is a nonnull cycle at x_2.*

Proof. We have $C = x_1 P$, where P is a path joining x_2 to x_1. Let $g = \Lambda(x_1, x_2)$. Then $\Lambda(C) = g.\Lambda(P)$ and $\Lambda(C') = \Lambda(P).g$. If we had $\Lambda(C') = 1_\Gamma$, then $g = \Lambda(P)^{-1}$, and we would obtain that $\Lambda(C) = 1_\Gamma$, impossible. □

We say that $\mathcal{G}$ is *consistent* iff it contains no nonnull cycle. We describe below a polynomial-time algorithm to verify consistency. Let $\lambda : V \to \Gamma$. We say that λ is a *consistent labelling* of $\mathcal{G}$ iff for each path P in $\mathcal{G}$ joining u to v, $\Lambda(P) = \lambda(u)^{-1}\lambda(v)$. Observe that if $\mathcal{G}$ has a consistent labelling, then it is consistent, for if C is a cycle at x in $\mathcal{G}$ we have $\Lambda(C) = \lambda(x)^{-1}\lambda(x) = 1_\Gamma$.

Lemma 8. *There is a polynomial-time algorithm which, given a Γ-labelled graph $\mathcal{G} = (V, A, \Lambda)$:*(a) *either finds a null cycle in $\mathcal{G}$ (and concludes that $\mathcal{G}$ is inconsistent);* (b) *or finds a consistent labelling of $\mathcal{G}$ (and concludes that $\mathcal{G}$ is consistent).*

Proof. The algorithm computes $F = (V, S)$ spanning forest of G. Then, in each connected component K of F, it chooses an arbitrary "root" $r_K \in K$. It then computes $\lambda : V \to \Gamma$ as follows: if a vertex v belongs to a connected component K of F, and if P_v is the unique path joining r_K to v in F, then $\lambda(v) := \Lambda(P)$. Such a labelling λ will be called an *F-consistent labelling of $\mathcal{G}$*.

Say that an arc $(u, v) \in A$ is *bad* iff $uv \notin S$ and $\Lambda(u, v) \neq \lambda(u)^{-1}\lambda(v)$. There are two cases, depending on the existence of a bad arc.

Case 1: there is a bad arc $a = (u, v)$ with u, v in a same connected component K of F. Then consider the cycle $C = uP_v^{-1}P_u$. We then have $\Lambda(C) = \Lambda(a)\Lambda(P_v)^{-1}\Lambda(P_u) = \Lambda(a)\lambda(v)^{-1}\lambda(u)$. Since $\Lambda(a) \neq \lambda(u)^{-1}\lambda(v)$, it follows that C is a nonnull cycle.

Case 2: there is no bad arc. We show that: for every arc $(u, v) \in A$, $\Lambda(u, v) = \lambda(u)^{-1}\lambda(v)$. This holds by assumption if $uv \notin S$. Suppose now that $uv \in S$ with u, v in a same connected component K of F. If $u \in P_v$, then $\lambda(v) = \Lambda(P_v) = \Lambda(P_u)\Lambda(u, v)$, which implies that $\Lambda(u, v) = \Lambda(P_u)^{-1}\Lambda(P_v) = \lambda(u)^{-1}\lambda(v)$. The case when $v \in P_u$ is symmetric. A straightforward induction on m then shows: if $P = x_1, \ldots, x_m$ is a path in $\mathcal{G}$, then $\Lambda(P) = \lambda(x_1)^{-1}\lambda(x_m)$. We conclude that λ is a consistent labelling of $\mathcal{G}$. □

It follows that consistency has a good characterization (in the sense of [13]): $\mathcal{G}$ has no nonnull cycle iff $\mathcal{G}$ has a consistent labelling.

4.2. *The* Group Feedback Vertex Set *problem*

Consider a Γ-labelled graph $\mathcal{G} = (V, A, \Lambda)$. If $S \subseteq V$, removing S from V produces the Γ-labelled graph $\mathcal{G} \setminus S = (V', A', \Lambda')$, where $V' = V - S$, $A' = \{(u, v) \in A : u, v \notin S\}$ and $\Lambda' = \Lambda|A'$. A *feedback vertex set* of $\mathcal{G}$ is a set $S \subseteq V$ s.t. $\mathcal{G} \setminus S$ is consistent (or equivalently, a set of vertices which meets every nonnull cycle). We consider the Group Feedback Vertex Set (GFVS) problem: given a Γ-labelled graph $\mathcal{G}$, a set $F \subseteq V$ of forbidden vertices, and an integer p, can we find a feedback vertex set of $\mathcal{G}$ disjoint from F and of size at most p? This section is devoted to the proof of the following theorem.

Theorem 5. GFVS *is solvable in $O^*((4|\Gamma| + 1)^p)$ time.*

The algorithm relies on iterative compression; see e.g. [3,14] for an introduction to the method. We introduce the GFVS Compression problem, which takes

1. a Γ-labelled graph $\mathcal{G} = (V, A, \Lambda)$,
2. a feedback vertex set S of $\mathcal{G}$,
3. a function $\phi : S \to \Gamma$,
4. a set $F \subseteq V$ of forbidden vertices,

and an integer p, and seeks a set S' of $< p$ vertices disjoint from F and which meets each path P joining two vertices $u, v \in S$ with $\Lambda(P) \neq \phi(u)^{-1}\phi(v)$.

The GFVS Compression problem will allow us to solve the compression step for GFVS, thanks to the following result.

Proposition 2. *Let $\mathcal{G} = (V, A, \Lambda)$, and let $F \subseteq V$. Let S be a feedback vertex set of $\mathcal{G}$. The following are equivalent:*

- *there exists a feedback vertex S' of $\mathcal{G}$ s.t. S' is disjoint from $F \cup S$ and $|S'| < |S|$;*
- *there exists $\phi : S \to \Gamma$ s.t. $I_\phi = (\mathcal{G}, S, \phi, F \cup S, |S|)$ is a positive instance of* GFVS Compression.

Proof. ($\Rightarrow$): suppose that there exists S' as stated. Since $\mathcal{G} \setminus S'$ is consistent, by Lemma 8 it admits a consistent labelling λ. Let ϕ be the restriction of λ to S. For each path P in $\mathcal{G} \setminus S'$ joining $u, v \in S$, we have $\Lambda(P) = \lambda(u)^{-1}\lambda(v) = \phi(u)^{-1}\phi(v)$, where the first equality follows from the fact that λ is a consistent labelling. We conclude that S' is a solution for GFVS Compression on the instance I_ϕ.

($\Leftarrow$): suppose that there exists $\phi : S \to \Gamma$ as stated. Let S' be a solution for GFVS Compression on the instance I_ϕ. Then S' is disjoint from $F \cup S$ and $|S'| < |S|$. We claim that $\mathcal{G} \setminus S'$ is consistent. Suppose by contradiction that C is a nonnull cycle of $\mathcal{G} \setminus S'$. Since S is a feedback vertex set of $\mathcal{G}$, it follows that C contains an element $x \in S$. Consider the cycle C' at x obtained as a circular permutation of C, then C' is nonnull by Lemma 7. Since S' is a solution for GFVS Compression, it follows that $\Lambda(C') = \phi(x)^{-1}\phi(x) = 1_\Gamma$, contradiction. We conclude that S' is a feedback vertex set of $\mathcal{G}$. □

By Proposition 2, an FPT algorithm for GFVS Compression will yield an algorithm for the GFVS problem.

Proposition 3. GFVS Compression *is solvable in $O^*(4^p)$ time.*

Proof. We formulate it as a path transversal problem for a homogeneous path system. Let $I = (\mathcal{G}, S, \phi, F, p)$ be an instance of GFVS Compression. We define the path system $\sigma = (G, S, F, \mathcal{P})$, where G is the underlying graph of $\mathcal{G}$, and where $\mathcal{P}$ consists of the paths P joining two vertices $u, v \in S$ with $\Lambda(P) \neq \phi(u)^{-1}\phi(v)$. We verify that σ is homogeneous and has a separation oracle:

(a) Point 1 of Definition 1: let P be a bad path which is inclusionwise minimal. Aiming for contradiction, suppose that P is not simple. Then $P = P_1CP_2$, where C is a cycle at $x \in V$. If C is null, then P_1xP_2 is a bad path included in P, impossible. If C is nonnull, it contains an element $y \in S$. Then the cycle C' at y obtained from C is also nonnull by Lemma 7, and is therefore a bad path included in P, impossible. We conclude that P is simple.

(b) Point 2 of Definition 1: let P be a bad path joining $u, v \in S$. Suppose that $P = P_1xP_2$, and let P' be a path joining some $w \in S$ to x. Aiming for contradiction, suppose that both $P_1P'^{-1}$ and $P'P_2$ are good paths. We then have $\Lambda(P_1P'^{-1}) = \phi(u)^{-1}\phi(w)$ and $\Lambda(P'P_2) = \phi(w)^{-1}\phi(v)$. But this implies that $\Lambda(P) = \Lambda(P_1P'^{-1})\Lambda(P'P_2) = \phi(u)^{-1}\phi(v)$, a contradiction.

(c) σ has a polynomial-time separation oracle: suppose that we are given $(d_v)_{v\in V}$. For each $u, v \in V, g \in \Gamma$, let $w(u, v, g)$ be the length of a shortest path P in G_d which joins u to v and satisfies $\Lambda(P) = g$. These values can be computed in $O(n^3|\Gamma|)$ time. A shortest bad path comes from a triple (u, v, g), with $u, v \in S$ and $g \neq \phi(u)^{-1}\phi(v)$, which minimizes $w(u, v, g)$.

We conclude by Theorem 1 that GFVS Compression can be solved in $O^*(4^p)$ time. □

We are now in position to prove Theorem 5.

Proof of Theorem 5. We solve Group Feedback Vertex Set using iterative compression. In the compression step, we are given a subset V' of V, a feedback set S of $\mathcal{G}[V']$ disjoint from F and of size p, and we seek S' feedback set of $\mathcal{G}[V']$ disjoint from F and of size $< p$. We examine every possibility for $S \cap S'$: for each bipartition of $S = S_1 \cup S_2$, we seek $S' = S_1 \cup S_2'$ with $S_2 \cap S_2' = \emptyset$ and $|S_2'| < |S_2|$. Let $i = |S_2|$, then finding S_2' amounts to find a feedback vertex set of $\mathcal{G} \setminus S_1$ disjoint from $F \cup S_2$ of size $< i$. This is done in $O^*(|\Gamma|^i \times 4^i)$ time: by Proposition 2, we need to examine the $|\Gamma|^i$ functions $\phi : S_2 \to \Gamma$, and for each such function we solve GFVS Compression in $O^*(4^i)$ time by Proposition 3. By summing on each possible value of i, we obtain that the total time required by the compression step is $\sum_{i=0}^{p} \binom{p}{i} O^*((4|\Gamma|)^i) = O^*((4|\Gamma| + 1)^p)$. Since there are at most n compression steps, the running time of the algorithm is as claimed. □

4.3. *The* Group Feedback Edge Set *problem*

We now consider the edge-version of the problem. Consider a Γ-labelled graph $\mathcal{G} = (V, A, \Lambda)$ with underlying graph $G = (V, E)$. If $S \subseteq E$, removing S from $\mathcal{G}$ produces the Γ-labelled graph $\mathcal{G} \setminus S = (V, A', \Lambda')$ where $A' = A - \{(u, v), (v, u) : uv \in S\}$, and $\Lambda' = \Lambda|A'$. A *feedback edge set* of $\mathcal{G}$ is a set $S \subseteq E$ s.t. $\mathcal{G} \setminus S$ is consistent. The Group Feedback Edge Set (GFES) problem asks: given a Γ-labelled graph $\mathcal{G}$ with underlying graph $G = (V, E)$, a set $F \subseteq E$ of forbidden edges, and an integer p, can we find a feedback edge set of $\mathcal{G}$ disjoint from F and of size at most p?

For the GFES problem, we obtain an algorithm with the same running time as for GFVS, as well as another algorithm whose exponential factor does not depend on $|\Gamma|$.

Theorem 6. GFES *is solvable in $O^*((4|\Gamma| + 1)^p)$ time, and in $O^*((8p + 1)^p)$ time.*

We rely on iterative compression in a similar fashion to the proof of Theorem 5. We define the GFES Compression problem, which takes

1. a Γ-labelled graph $\mathcal{G} = (V, A, \Lambda)$ with underlying graph $G = (V, E)$,
2. a feedback vertex set S of $\mathcal{G}$,
3. a function $\phi : S \to \Gamma$,
4. a set $F \subseteq E$ of forbidden edges,

and an integer p, and seeks a set S' of $< p$ edges disjoint from F and which meets each path P joining two vertices $u, v \in S$ with $\Lambda(P) \neq \phi(u)^{-1}\phi(v)$.

The following proposition demonstrates how the GFES Compression problem serves in the compression step. Its proof is similar to Proposition 2 and is therefore omitted.

Proposition 4. *Let $\mathcal{G} = (V, A, \Lambda)$ with underlying graph $G = (V, E)$, and let $F \subseteq E$. Let $S \subseteq E$ be a feedback edge set of $\mathcal{G}$, and let $K \subseteq V$ be a vertex cover of S. The following are equivalent:*

- *there exists a feedback edge set S' of $\mathcal{G}$ s.t. S' is disjoint from $F \cup S$ and $|S'| < |S|$;*
- *there exists $\phi : K \to \Gamma$ s.t. $(\mathcal{G}, K, \phi, F \cup S, |S|)$ is a positive instance of* GFES Compression.

This will yield an algorithm for the compression step of GFES, provided that we have an algorithm for the GFES Compression problem. We now present such an algorithm.

Proposition 5. GFES Compression *is solvable in $O^*(4^p)$ time.*

Proof. We describe a simple reduction to GFVS Compression, and conclude using Proposition 3.

Let $I = (\mathcal{G}, S, \phi, F, p)$ be an instance of GFES Compression, where $\mathcal{G} = (V, A, \Lambda)$ is a Γ-labelled graph with underlying graph $G = (V, E)$. We create an instance $I' = (\mathcal{G}', S, \phi, F', p)$ of GFVS Compression as follows. For each edge $e = uv \in E$, we introduce a new vertex x_e; then, if the arcs $(u, v), (v, u)$ have labels g, g^{-1}, we replace them by arcs $(u, x_e), (x_e, v), (v, x_e), (x_e, u)$ having respective labels $g, 1_\Gamma, 1_\Gamma, g^{-1}$. This defines $\mathcal{G}' = (V', A', \Lambda')$ which is easily seen to be a Γ-labelled graph. The set of forbidden vertices is $F' = V \cup \{x_e : e \in F\}$.

The correctness of the reduction follows by observing that the function which maps $S \subseteq E$ to $S' = \{x_e : e \in S\}$ yields a bijection between the solutions of I' and of I. □

Let us now prove Theorem 6. The $O^*((4|\Gamma| + 1)^p)$ running time is obtained using iterative compression similarly to Theorem 5, with the difference that we now proceed by inserting edges. This is correct since if S is a feedback edge set of $\mathcal{G}$ and if $T \subseteq E$, then $S \setminus T$ is a feedback edge set of $\mathcal{G} \setminus T$. The running time analysis is similar to Theorem 5, except that the number of compression steps is now $\leq |E|$. Note that when applying Proposition 4 to S solution of size p, we can find K vertex cover of S of size p, therefore the number of functions ϕ to examine is $|\Gamma|^p$, as in Theorem 5.

We now argue that the number of functions ϕ to consider can be reduced from $|\Gamma|^p$ to $(2p)^p$, which will yield an $O^*((8p + 1)^p)$ algorithm. We introduce the following definitions. For each connected component C of G s.t. $C \cap K \neq \emptyset$, choose an element $r_C \in C \cap K$. Let K_1 be the set of chosen elements of K. Say that a function $\phi : K \to \Gamma$ is *canonical* iff $\phi(x) = 1_\Gamma$ for each $x \in K_1$. Say that a function $\phi : K \to \Gamma$ is *realizable* iff it is canonical and there exists a spanning forest F of G and an F-consistent labelling λ of $\mathcal{G}$ s.t. $\phi = \lambda|K$. Observe that Proposition 4 remains correct if we require ϕ to be realizable.

The following lemma gives an upper bound on the number of realizable functions.

Lemma 9. *The number of realizable functions is at most* $(2p)^p$.

Proof. We assume that K is the set of endpoints of edges in S. We say that an *S-forest* is a pair $\mathcal{F} = (S', F, \psi, \psi')$, consisting of: (a) a set $S' \subseteq S$, (b) a forest $F = (U, E)$ with $|S'|$ edges, (c) a function $\psi : K \to U$, (d) a bijective function $\psi' : S' \to E$, satisfying the two following properties: (i) for every $x \in U$, $\psi^{-1}(x)$ is included in a connected component of $G \setminus S$, (ii) for $e = uv \in S'$, it holds that $\psi'(e) = \psi(u)\psi(v)$.

To an S-forest $\mathcal{F}$, we associate a function $\phi_\mathcal{F} : K \to \Gamma$ as follows. Suppose that $u \in K$ belongs to a connected component C of G. Let $P = x_1, \ldots, x_m$ be the path joining $\psi(r_C)$ to $\psi(u)$ in F. Then there exists vertices $u_1v_1, \ldots, u_mv_m$ in K s.t.

(a) $u_1 = r_C$, $v_m = u$,
(b) for every $1 \leq i \leq m$, there is a path P_i from u_i to v_i in $G \setminus S$ (by (i)),
(c) for every $1 \leq i < m$, it holds that $v_iu_{i+1} \in S'$.

Let P_u be the concatenation of the paths $P_1, (v_1, u_2), \ldots, (v_{m-1}, u_m), P_m$. We then define $\phi_\mathcal{F}(u) = \Lambda(P_u)$.

We claim that every realizable function is obtained by this construction. Indeed, suppose that $\phi : K \to \Lambda$ is realizable, let F be the corresponding spanning forest of G and λ be the F-consistent labelling s.t. $\phi = \lambda|K$. Let F' be obtained from F by contracting the edges of $E \setminus S$. Then each edge $e \in F'$ corresponds to an edge $\sigma_e \in S$, and each vertex u of F' corresponds to a set of vertices V_u of G. Besides, for each edge $e = uv \in F'$, if $\sigma_e = xy$ then $x \in V_u, y \in V_v$. It follows that if we define $\mathcal{F} = (S', F', \psi, \psi')$ with

1. for each $x \in K$, $\psi(u)$ being the unique $u \in V(F')$ s.t. $x \in V_u$,
2. for each $f \in S$, $\psi'(f)$ being the unique $e \in E(F')$ s.t. $f = \sigma_e$ (if it exists),
3. S' being the definition domain of ψ',

then $\phi = \phi_{\mathcal{F}'}$. This equality holds because when computing $\phi_\mathcal{F}$ as above, the value $\Lambda(P_i)$ does not depend on a particular choice of P_i (since $\mathcal{G} \setminus S$ is consistent).

We conclude that the number of realizable functions is upper bounded by the number of S-forests, which is at most $(2p)^p$. □

From the proof of the lemma, we obtain an algorithm which enumerates the realizable functions in $O^*((2p)^p)$ time.

5. Concluding remarks

The parameterized complexity of some problems considered in this article remains unsettled. We have seen that the Vertex Multicut and Multicut problems were FPT w.r.t. k, p, and that the Group Feedback Vertex Set problem was FPT w.r.t. $|\Gamma|, p$. It is an open question whether these problems are FPT for the single parameter p. Note that in the case of Multicut it was first mentioned by [4]. Also, the existence of polynomial kernels for these problems is open, as well as for Multiway Cut. Could the half-integrality property be useful in this respect, as it is for the kernelization of Vertex Cover?

We also think that some variants of the problems on group-labelled graphs considered in Section 4 may be fixed-parameter tractable. An interesting generalization of the Group Feedback Edge Set problem is the Unique Label Cover problem, whose approximability was studied in connection with the Unique Games Conjecture [15,16]. Other problems of interest are satisfiability problems for systems of linear equations/inequations, parameterized by the maximum number of unsatisfied equations allowed. These problems may be FPT when restricted to instances with at most two variables per equation.

More generally, could our LP-based approach for Path Transversal apply to other problems? It may prove useful in other cases, possibly in combination with iterative compression.

References

[1] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer-Verlag, 1999.
[2] J. Flum, M. Grohe, Parameterized Complexity Theory, Springer-Verlag, 2006.
[3] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006.
[4] D. Màrx, Parameterized graph separation problems, Theoret. Comput. Sci. 351 (2006) 394–406.
[5] J. Chen, Y. Liu, S. Lu, An improved parameterized algorithm for the minimum node multiway cut problem, Algorithmica 55 (1) (2009) 1–13.
[6] M. Xiao, Algorithms for multiterminal cuts, in: Proc. CSR 2008, in: Lecture Notes in Computer Science, vol. 5010, 2008, pp. 314–325.
[7] J. Chen, F.V. Fomin, Y. Liu, S. Lu, Y. Villanger, Improved algorithms for feedback vertex set problems, J. Comput. System Sci. 74 (7) (2008) 1188–1198.
[8] B. Reed, K. Smith, A. Vetta, Finding odd cycle transversals, Oper. Res. Lett. 32 (4) (2004) 299–301.
[9] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, I. Razgon, A fixed-parameter algorithm for the directed feedback vertex set problem, J. ACM 55 (5) (2008).
[10] M. Chudnovsky, J. Geelen, B. Gerards, L.A. Goddyn, M. Lohman, P.D. Seymour, Packing non-zero A-paths in group-labelled graphs, Combinatorica 5 (26) (2006) 521–532.
[11] K.I. Kawarabayashi, P. Wollan, Non-zero disjoint cycles in highly connected group labeled graphs, J. Combin. Theory Ser. B 96 (2006) 296–301.
[12] N. Garg, V. Vazirani, M. Yannakakis, Multiway cuts in directed and node weighted graphs, in: Proc. ICALP'94, in: Lecture Notes in Computer Science, vol. 820, 1994, pp. 487–498.
[13] J. Edmonds, Path, trees and flowers, Canad. J. Math. 17 (1965) 449–467.
[14] J. Guo, H. Moser, R. Niedermeier, Iterative compression for exactly solving NP-hard minimization problems, in: Algorithmics of Large and Complex Networks, in: Lecture Notes in Computer Science, vol. 5515, 2009, pp. 65–80.
[15] S. Khot, On the power of unique 2-prover 1-round games, in: Proc. STOC'02, 2002, pp. 767–775.
[16] A. Gupta, K. Talwar, Approximating unique games, in: Proc. SODA'06, 2006, pp. 99–106.