LP-branching algorithms based on biased graphs

Magnus Wahlström *

Dept of Computer Science,
Royal Holloway, University of London,
UK

October 20, 2016

Abstract

We give a combinatorial condition for the existence of efficient, LP-based FPT algorithms for a broad class of graph-theoretical optimisation problems. Our condition is based on the notion of biased graphs known from matroid theory. Specifically, we show that given a biased graph $\Psi = (G, \mathcal{B})$, where $\mathcal{B}$ is a class of balanced cycles in G, the problem of finding a set X of at most k vertices in G which intersects every unbalanced cycle in G admits an FPT algorithm using an LP-branching approach, similar to those previously seen for VCSP problems (Wahlström, SODA 2014). Our algorithm has two parts. First we define a *local problem*, where we are additionally given a root vertex $v_0 \in V$ and asked only to delete vertices X (excluding v_0) so that the connected component of v_0 in $G - X$ contains no unbalanced cycle. We show that this local problem admits a persistent, half-integral LP-relaxation with a polynomial-time solvable separation oracle, and can therefore be solved in FPT time via LP-branching, assuming only oracle membership queries for the class of balanced cycles in G. We then show that solutions to this local problem can be used to tile the graph, producing an optimal solution to the original, global problem as well. This framework captures many of the problems previously solved via the VCSP approach to LP-branching, as well as new generalisations, such as Group Feedback Vertex Set for infinite groups (e.g., for graphs whose edges are labelled by matrices). A major advantage compared to previous work is that it is immediate to check the applicability of the result for a given problem, whereas testing applicability of the VCSP approach for a specific VCSP, requires determining the existence of an embedding language with certain algebraically defined properties, which is not known to be decidable in general.

1 Introduction

In recent years, we have seen a growing interest in the use of linear or integer linear programming methods (LP/ILP) in parameterized complexity [14, 15, 10, 16]. The appeal is clear. On the one hand, linear programming and general continuous relaxations comes with a very powerful toolbox for theoretical investigation. This promises to be a powerful hammer, especially for optimisation problems; e.g., an FPT-size extended formulation, or more generally an FPT separation oracle for an integral polytope would provide a way towards FPT algorithms [16]. The same can be said for Lenstra's algorithm and other methods for solving complex problems in few variables [18, 3]. On the other hand, it has been observed that ILP solvers, like SAT solvers, frequently perform much better in practice than can be currently be explained theoretically. It

*Magnus.Wahlstrom@rhul.ac.uk

is appealing to study parameterizations of ILP problems, to find structural reasons for this apparent tractability [15, 10].

Narrowing our focus, for many optimisation problems we have seen powerful FPT results based on running an ILP-solver with performance guarantees. These algorithms use LP-relaxations with strong structural properties (*persistence* and *half-integrality*), which allows you to run an FPT branching algorithm on top of the relaxation, typically getting a running time of $O^*(2^{O(k)})$ for a solution size of k. More powerfully, these algorithms also allow to bound the running time in terms of the *relaxation gap*, i.e., the additive difference between the value of the LP-optimum and the true integral optimum. For problems such as Vertex Cover [19] and Multiway Cut [6], this gives us FPT algorithms that find an optimum in time $O^*(2^{O(k-\lambda)})$, where k is the size of the optimum and λ is the lower bound given by the LP-relaxation of the input. In this sense, these results can also be taken as a possible "parameterized explanation" of the success of ILP solvers for these problems, although this approach of course only works for certain problems as it is in general NP-hard to decide whether an ILP problem has a solution matching the value of the LP-relaxation.

The main restriction for this approach is to find LP-relaxations with the required structural properties (or even to find problems which admit such LP-relaxations). In the cases listed above, this falls back on classical results from approximation [22, 11], but further examples proved elusive. A significant advancement was made recently [14] by recasting the search for LP-relaxations with the required properties in algebraic terms, using a connection between such LP-relaxations and so-called *Valued CSPs* (VCSPs; see below). Using this connection, FPT LP-branching algorithms were provided that significantly improved the running times for a range of problems, including Subset Feedback Vertex Set and Group Feedback Vertex Set in $O^*(4^k)$ time for solution size k, improving on previous records of $O^*(2^{O(k \log k)})$ time [7, 5], and Unique Label Cover in $O^*(|\Sigma|^{2k})$ time for alphabet Σ and solution size k, improving on a previous result of $O^*(|\Sigma|^{O(k^2 \log k)})$ [1]. Group Feedback Vertex Set in particular is a meta-problem that includes many independently studied problems as special cases.

However, powerful though these results may be, the nature of the framework makes it difficult to apply to a given combinatorial problem. To do so would involve two steps. First, the problem must be phrased as a VCSP. A VCSP asks to minimise the value of an objective function $f(\phi)$ over assignments $\phi : V \to D$ from a finite domain D, where the objective $f(\phi)$ is in turn usually given as a sum of bounded-size *cost functions* $\sum_{i=1}^{m} f_i(\phi)$. To find such a formulation is not always easy, and sometimes it may be impossible to capture a problem precisely. As an example, it is possible to phrase Feedback Vertex Set as an instance of Group Feedback Vertex Set and hence arguably as a VCSP [14], but to discover such a formulation from first principles is not easy. Second, given a desired VCSP formulation, it must be determined whether the VCSP admits a *discrete relaxation*, so that the framework can be applied [14]. It is not known in general how to decide the existence of such a relaxation.[1]

In short, although the results are powerful, they are also somewhat inscrutable.

In this paper, we instead give a combinatorial condition under which a graph-theoretical problem admits an LP-relaxation with the required properties, and hence an efficient FPT algorithm parameterized either by solution size, or in particular cases by a relaxation gap parameter. Our condition is based on the class of so-called *biased graphs*, which are combinatorial objects of importance especially to matroid theory [28, 29]. We review these next.

[1] The results of [14] are obtained by working backwards from known relaxations with the required properties, a list which includes separable k-submodular relaxations and arbitrary bisubmodular relaxations, and with a small extension can be made to cover a class of problems also including so-called *skew bisubmodular* functions [13]. However, again, even given a specific target class it is usually at least intuitively non-obvious whether a VCSP can be relaxed into the class or not.

Biased graphs. A *biased graph* is a pair $\Psi = (G = (V, E), \mathcal{B})$ of a graph G and a set $\mathcal{B} \subseteq 2^E$ of simple cycles in G, referred to as the *balanced cycles* of G. with the property that if two cycles $C, C' \in \mathcal{B}$ form a *theta graph* (i.e., a collection of three internally vertex-disjoint paths with shared endpoints), then the third cycle of $C \cup C'$ is also contained in $\mathcal{B}$. A cycle class $\mathcal{B}$ with this property is referred to as *linear*. Dually, and more important to the present paper, a simple cycle C is *unbalanced* if $C \notin \mathcal{B}$. The definition is equivalent to saying that if C is an unbalanced cycle, and if P is a path with endpoints in C which is internally edge- and vertex-disjoint from C, then at least one of the two new cycles formed by $C \cup P$ is also unbalanced. We refer to a collection $\mathcal{C}$ of cycles as a *co-linear* cycle class if the complement of $\mathcal{C}$ is a linear class. We say that an induced subgraph $G[S]$ of G is balanced if $G[S]$ contains no unbalanced cycles.

The basic problem considered in this paper is now defined as follows: Given a biased graph $\Psi = (G = (V, E), \mathcal{B})$ and an integer k, find a set of vertices $X \subseteq V$ with $|X| \leq k$ such that $G - X$ is balanced. We refer to this as the Biased Graph Cleaning problem. Our main result in this paper is that Biased Graph Cleaning is FPT by k, with a running time of $O^*(4^k)$, assuming only access to a membership oracle for the class $\mathcal{B}$ (i.e., for a cycle C, given as a set of edges, we can determine whether $C \in \mathcal{B}$ with an oracle call).

An important example of biased graphs are *group-labelled graphs*. Let $G = (V, E)$ be an oriented graph, and let the edges of G be labelled by elements from a group $\Gamma = (D, \cdot)$, such that if an edge $uv \in E$ has label $\gamma \in D$, then the edge vu (i.e., uv traversed in the opposite direction) has label γ^{-1}. Then the balanced cycles of G are the cycles C such that the product of the edge labels of the cycle, read in the direction of their traversal, is equal to the identity element 1_Γ of Γ. Note that the orientation of the edges serves only to make the group-labelling well-defined, and has no bearing on which cycles we consider. It is easy to verify that this defines a linear class of cycles, and hence gives rise to a biased graph Ψ.

The problem Group Feedback Vertex Set corresponds exactly to Biased Graph Cleaning when Ψ is defined by a group-labelled graph. However, not all biased graphs can be defined via group labels, and moreover, some group-labelled graphs can only be defined via an infinite group Γ [8]. More examples of biased graphs follow below.

Biased graphs were originally defined in the context of matroid theory. Although this connection is not important to the present paper, we nevertheless give a brief review. Each biased graph Ψ gives rise to two matroids, the *frame matroid* and the *lift matroid* of Ψ. These are important examples in structural matroid theory. The *Dowling geometries* $Q_n(\Gamma)$ for a group Γ, originally defined by Dowling [9], are equivalent to frame matroids of complete Γ-labelled multigraphs. For more on matroids for biased graphs, see Zaslavsky [28, 29], as well as the series of blog posts on the *Matroid Union* weblog [23, 24, 25].

Our approach. Inspired by the previous algorithm for Group Feedback Vertex Set [14], our approach for Biased Graph Cleaning consists of two parts, the *local problem* and the *global problem*. In the local problem, the input is a biased graph $\Psi = (G = (V, E), \mathcal{B})$ together with a *root vertex* $v_0 \in V$ and an integer k, and the task is restricted to finding a set $X \subseteq V$ of vertices, $|X| \leq k$ and $v_0 \notin X$, such that the connected component of v_0 in $G - X$ is balanced. Equivalently, the local problem can be defined as finding a set $S \subseteq V$ with $v_0 \in S$ such that $G[S]$ is balanced and connected, and $|N_G(S)| \leq k$. We refer to this local problem as Rooted Biased Graph Cleaning.

We show that Rooted Biased Graph Cleaning can be solved via an LP which is half-integral and has a stability property similar to persistence. This LP uses a formulation where the constraints correspond to rooted cycles we refer to as *balloons*. The formulation of this slightly unusual LP is critical to the tractability of the problem. The possibly more natural approach of letting the obstacles of the LP simply be unbalanced cycles would not work as well;

for instance, although Feedback Vertex Set is an instance of Biased Graph Cleaning (with balanced cycles $\mathcal{B} = \emptyset$), it is known that the natural cycle-hitting LP has an integrality gap of a factor of $\Theta(\log n)$ [2].[2]

The properties of the LP further imply that Rooted Biased Graph Cleaning has a 2-approximation, even for weighted instances, and can be solved (in the unlighted case) in time $O^*(4^{k-\lambda})$ where λ is the value of the LP-optimum, assuming access to a membership oracle for the class $\mathcal{B}$ as above. In particular, Rooted Biased Graph Cleaning can be solved in time $O^*(2^k)$. We note that several independently studied problems arise as special cases of Rooted Biased Graph Cleaning; see below.

In order to solve the global problem, Biased Graph Cleaning, we show that the local LP obeys a strong persistence-like property, analogous to the *important separator* property frequently used in graph separation problems [20], which allows us to identify "furthest-reaching" local connected components when solving the local problem, such that the connected components produced by the algorithm for the local problem can be used to "tile" the original graph in a solution to the global problem. A $O^*(4^k)$-time algorithm for Biased Graph Cleaning follows (although the "above lower bound" perspective does not carry over to solutions for the global problem).

Results and applications. We summarise the above statements in the following theorems. Let $\Psi = (G = (V, E), \mathcal{B})$ be a biased graph, where $\mathcal{B}$ is defined via a membership oracle that takes as input a simple cycle C, provided as an edge set, and tests whether $C \in \mathcal{B}$. Then the following apply.

Theorem 1. *Assuming a polynomial-time membership oracle for the class of balanced cycles,* Rooted Biased Graph Cleaning *admits the following algorithmic results:*

- *A polynomial-time 2-approximation;*
- *An FPT algorithm with a running time of $O^*(4^{k-\lambda})$, where $\lambda \geq k/2$ is the value of the LP-relaxation of the problem;*
- *An FPT algorithm with a running time of $O^*(2^k)$.*

The 2-approximation holds even for weighted graphs.

Theorem 2. *Assuming a polynomial-time membership oracle for the class of balanced cycles,* Biased Graph Cleaning *admits an FPT algorithm with a running time of $O^*(4^k)$.*

To illustrate the flexibility of the notion, let us consider some classes of biased graphs.

- If $\mathcal{B} = \emptyset$, then Biased Graph Cleaning corresponds simply to Feedback Vertex Set
- If Ψ arises as a Γ-labelled graph, then Biased Graph Cleaning corresponds to Group Feedback Vertex Set. If Γ is finite, then the result is equivalent to the previous LP-based algorithm [14].

[2] Although formulations of LPs for Feedback Vertex Set exist with an integrality gap of 2 [2], we are not aware of any similarities between these formulations and our local LP. Moreover, other special cases of Biased Graph Cleaning, e.g., Odd Cycle Transversal, admit no constant-factor approximation unless the Unique Games Conjecture fails.

- If Ψ is Γ-labelled for an infinite group Γ, e.g., a matrix group, then previous results do not apply, since they assume the existence of an underlying VCSP presentation (in the correctness proofs, if not in the algorithms). However, the problem is still FPT, assuming, essentially, that the word problem for Γ can be solved.

- To show a case that does not obviously correspond to group-labelled graphs, let G be (improperly) edge-coloured, and let a cycle be balanced if and only if it is monochromatic. It is not difficult to see that this defines a biased graph.

- Finally, Zaslavsky [28] notes another case that in general does not admit a group-labelled representation. Let G be a copy of C_n, with two parallel copies of every edge. Let $\mathcal{B}$ be a class of "isolated" Hamiltonian cycles of G, in the sense that for any $C \in \mathcal{B}$, switching the edge used between u and v for any pair of consecutive vertices of G results in an unbalanced cycle. It is not hard to verify that this defines a biased graph. To keep G as a simple graph, we simply subdivide the edges; this does not affect the collection of cycles (although it means that the term "Hamiltonian" fails to apply).

  However, the corresponding problem can be very difficult, e.g., $\mathcal{B}$ may consist of only exactly one of the 2^n candidate cycles, giving an oracle lower bound of $\Omega(2^k)$ against any algorithm for Biased Graph Cleaning.

Finally, we show as promised that the algorithm for Rooted Biased Graph Cleaning has independent applications.

- Let $G = (V, E)$ be any graph, and add an apex vertex v_0 to G. Let $\mathcal{B} = \emptyset$. Then Rooted Biased Graph Cleaning corresponds to Vertex Cover, hence Theorem 1 is an $O^*(4^k)$-time FPT algorithm for the problem Vertex Cover Above Matching, which encompasses the more commonly known problem Almost 2-SAT [21, 26].

- Let $G = (V, E)$ be a graph and $T \subseteq V$ a set of terminals. Duplicate each terminal $t \in T$ into $d(t)$ copies, forming a set T', and add a vertex v_0 to G with $N(v_0) = T'$. Let a cycle be *unbalanced* if and only if it passes through v_0 and two vertices $t, t' \in T'$ which are copies of distinct terminals in T. Then Rooted Biased Graph Cleaning corresponds to Multiway Cut, and Theorem 1 reproduces the known 2-approximation and $O^*(2^k)$-time algorithm [11, 6].

  We note that the linear class condition prevents us from representing any other cut problem this way; if terminal-terminal paths from t to t' and from t' to t'' are allowed in G, for some $t, t', t'' \in T$, then also the path from t to t'' must be allowed. Hence the set of allowed paths induces an equivalence relation on T.

In general, we find that the notion of biased graphs is surprisingly subtle, and corresponds surprisingly well to the class of (natural) problems for which LP-branching FPT algorithms are known. We also note, without further study at the moment, that there is significant similarity between the (local or global) problems that can be expressed via biased graphs this way and the class of graph separation problems for which the existence of *polynomial kernels* is either known, or most notoriously open [17].

Finally, although our results do include some cases which were not previously known to be FPT, we feel that the most significant advantage of the present work is the transparency and naturalness of the definition. The existence of a purely combinatorial condition also arguably brings us closer to the existence of a purely combinatorial algorithm for these problems. Since the results in this paper rely on using an LP-solver with a separation oracle, a purely combinatorial algorithm could significantly decrease the hidden polynomial factor in the running times.

Preliminaries. We assume familiarity with the basic notions of graph theory, parameterized complexity, and the basics of combinatorial optimisation. For a reference on parameterized complexity, see Cygan et al. [4]; for all necessary material on linear programming and combinatorial optimisation, see Schrijver [27]. Other notions will be introduced as they are used.

2 Biased graphs and the local LP

In this section, we define the local LP, used to solve the local problem, and give some results about the structure of min-weight obstacles in it. We will also show that we can optimise over the LP in polynomial time by providing a separation oracle. In subsequent sections, we will derive the properties of half-integrality and persistence, and show Theorems 1 and 2.

We first introduce some additional terminology. Recall the definition of biased graphs from Section 1. Let $\Psi = (G = (V, E), \mathcal{B})$ be a biased graph. For a simple cycle C, a *chord path* for C is a simple path with end vertices in C and internal vertices and edges disjoint from C. If C is an unbalanced cycle in Ψ, a *reconfiguration of C by P* refers to an unbalanced cycle C' formed from C and P, with C' containing P, as is guaranteed by the definition of biased graphs. Note that a chord path can consist of a single edge and no internal vertices; however, it is also possible that a chord path is non-induced, e.g., for structural purposes we may reconfigure a cycle C using a chord path P that contains internal vertices, even if there is a direct edge in G connecting the end points of P. A *membership oracle* for $\mathcal{B}$ is a (black-box) algorithm which, for every set of edges forming a simple cycle C in G, will respond whether $C \in \mathcal{B}$.

As a warm-up, and to illustrate the kind of arguments we will be using, we give a simple lemma that shows why co-linearity is a useful structural property from the perspective of the local problem.

Lemma 1. *Let $\Psi = (G = (V, E), \mathcal{B})$ be a biased graph, and let $V_R \subseteq V$ be a set of vertices such that $G[V_R]$ is balanced and connected. Let C be an unbalanced cycle. Then either C intersects $N(V_R)$ in at most one vertex, or there exists an unbalanced cycle C' such that C' intersects V_R in a non-empty simple path, C' intersects $N(V_R)$ in at most two vertices, and $(V(C') \setminus V_R) \subseteq (V(C) \setminus V_R)$.*

Proof. Assume that C intersects $N(V_R)$ at least twice, as otherwise there is nothing to show. Assume first that C is disjoint from V_R, and let u and v be two distinct members of $V(C) \cap N(V_R)$. Let P be a uv-path with internal vertices in V_R. Reconfiguring C using P as a chord path results in an unbalanced cycle C' which intersects V_R, and whose intersection with $V \setminus V_R$ is a subset of that of C, hence we may assume that the cycle C intersects V_R.

Now consider $V(C) \cap V_R$. Since V_R is balanced, this is a collection of paths (rather than the entire cycle). Let P_C be one such path, and let u and w be the vertices in $N(V_R)$ that it terminates at. Now, if possible, let P be a shortest path in V_R connecting P_C to a vertex of either $(N(V_R) \cap V(C)) \setminus \{u, w\}$ or $(V(C) \setminus V(P_C)) \cap V_R$. In both cases, using P as a chord path results in a new unbalanced cycle C' whose intersection with $N(V_R)$ is strictly decreased.

The only remaining case is that $V(C) \cap V_R$ forms a single path, and $V(C) \cap N(V_R)$ consists of exactly two vertices (necessarily the attachment points of the path), and we are done. □

Properties and arguments similar to this will be used extensively in the arguments concerning the behaviour of the LP.

2.1 The LP relaxation

We now define the LP-relaxation used for the Rooted Biased Graph Cleaning. The LP uses constraints we refer to as *balloons*; we will see that balloons can equivalently be thought of as pairs of paths rooted in v_0, or as a cycle connected to v_0 by a path.

Let $\Psi = (G = (V, E), \mathcal{B})$ be a biased graph and $v_0 \in V$ a distinguished vertex. Let $\mathcal{C}$ be the corresponding class of unbalanced simple cycles. The Rooted Biased Graph Cleaning problem asks for a set $S \subseteq V$ with $v_0 \in S$ such that $G[S]$ is balanced and connected, and $|N(S)|$ is minimum. We consider the following LP-relaxation for it. The variables are $\{x_v : v \in V\}$, with $0 \leq x_v \leq 1$ and $x_{v_0} = 0$. For $C \in \mathcal{C}$, a *v_0-C-path* is a simple path $P = v_0 \dots v_\ell$ where $v_\ell \in C$ and $v_i \notin C$ for $1 \leq i < \ell$. If $v_0 \in C$, then P consists of the single vertex v_0 and no edges. We define the weights of C and P as $w(C) = \sum_{v \in C} x_v$ and $w(P) = \sum_{v \in P - v_\ell} x_v + \frac{1}{2} x_{v_\ell}$, i.e., $w(P)$ assigns coefficient $\frac{1}{2}$ to the endpoint of P and 1 to the internal vertices of P. A *(v_0-)balloon* is a pair $B = (P, C)$ where $C \in \mathcal{C}$ and P is a v_0-C-path; the *weight* of B is $w(B) = 2w(P) + w(C)$. We call the endpoint v_ℓ of P the *knot vertex* of B. We let $V(B)$ (respectively $V(C)$, $V(P)$) denote the set of vertices occurring in B (respectively in C, in P); hence $V(C) \cap V(P) = \{v_\ell\}$ is the knot vertex. The *edges used in* B, $E(B)$, is the set of edges required for B to be a balloon, i.e., the edges $v_i v_{i+1}$ of P and the edges of C. Note that this does not necessarily include all edges of $G[V(B)]$. The *edges used in* P (in C) is defined correspondingly.

Having fixed v_0 and $\mathcal{C}$ as above, we define a polytope $\mathcal{P}$ by constraints

$$w(B) \geq 1 \text{ for every } v_0\text{-balloon } B = (P, C).$$

with $x_v \geq 0$ for every vertex v and $x_{v_0} = 0$. We refer to this as the *local LP*.

Given an optimisation goal $\min c^T x$ for the above LP, the dual of the system asks to pack balloons, at weight 1 for every balloon, subject to every vertex v having a capacity c_v and a balloon $B = (P, C)$ using capacity from v in proportion to the coefficient of v in $w(B)$ (which is 2 if $v \in V(P)$, and 1 otherwise).

2.2 Balloons and path pairs

We now give some observations that will simplify the future arguments regarding the local LP, and in particular will allow us to change perspectives between viewing the constraints as pairs of paths, or as rooted unbalanced cycles.

Let $x : V \to [0, 1]$ be a fractional assignment. We first observe that our weights $w(P)$ and $w(C)$ for balloons $B = (P, C)$ can be recast as *edge weights*: For any edge $uv \in E$, we let the length of uv under x be $\ell_x(uv) = (x_u + x_v)/2$. For a path P, we let $\ell_x(P) = \sum_{uv \in E(P)} \ell_x(uv)$ be the length of P under this metric, and similarly for simple cycles. We also define $z_x(v) = \min_P \ell_x(P)$ ranging over all v_0-v-paths P as the *distance to v (under x, from v_0)*. This metric agrees well with the notion of the weight of a balloon that we use in the LP, as we will see. Note that the end points of a path P contribute only half their weight to the length $\ell_x(P)$ of a path, as in $w(P)$.

Now observe that for any balloon $B = (P, C)$, and any vertex $v \in C$ other than the knot vertex, it is possible to form two paths P_1, P_2 from v_0 to v, such that $P_1 \cup P_2$ covers B, and such that the weight of the balloon equals $\ell_x(P_1) + \ell_x(P_2)$. Indeed, this sum gives coefficient 2 to every vertex of P, and coefficient 1 to every vertex of $V(C) \setminus V(P)$, including the vertex v. We refer to this as a path decomposition of B. We also observe two alternative decompositions.

Lemma 2. *Let $B = (V, E)$ be a v_0-balloon. Each of the following is an equivalent decomposition of the weight of B.*

1. $w(B) \geq \ell_x(C) + \min_{v \in V(C)} 2z_x(v)$*, with equality achieved if B is a min-weight v_0-balloon.*
2. $w(B) = \ell_x(P_1) + \ell_x(P_2)$*, for any decomposition of B into two paths P_1, P_2 ending at a non-knot vertex $v \in V(C)$.*

3. $w(B) = \ell_x(P_u)+\ell_x(P_v)+\ell_x(uv)$ *for any decomposition of* B *into one path* P_u *to* $u \in V(C)$, *one path* P_v *to* $v \in V(C)$, *and an edge* $uv \in E(C)$, *where* $u, v \in V(C)$ *are non-knot vertices. Equivalently,* $w(B) = \sum_{w \in P_u} x_w + \sum_{w \in P_v} x_w$.

Note that the first decomposition here implies that the knot vertex v of a min-weight balloon B will be chosen for minimum z_x-value.

In general, we view constraints as pairs of paths when deriving simple properties of the LP, but will need to revert to the view of biased graphs when arguing persistence in the next section.

2.3 Structure of min-weight balloons

We now work closer towards a separation oracle, by showing properties of balloons B minimising $w(B)$ under an assignment x, as this will help us finding the *most violated constraint* of an instance of the local LP.

Our first lemma is a structural result that will be independently useful in the next section.

Lemma 3. *Let* $B = (P, C)$ *be a min-weight balloon with respect to an assignment* $x : V \to [0, 1]$. *Then for every vertex* v *in* B, B *contains a shortest* v_0-v*-path under the metric* ℓ_x.

Proof. Clearly, P must be a shortest path, by decomposition 1 of Lemma 2, hence the claim holds for every vertex of P. Let $v \in C$, and let P_v be a shortest v_0-v-path such that both paths to v along B (as in decomposition 2) are longer than P_v; let v be chosen with minimum $z_x(v)$-value, subject to these conditions. We may choose P_v so that $P_v \cap P$ is a prefix of both. We may also assume that P_v "rejoins" B only exactly once, i.e., after P_v has followed its first edge not present in B, then no further edge of P_v is contained in B. The reason for this assumption is that every vertex of $V(P_v) \cap V(B)$ prior to v has a shortest path contained in B, by choice of v. Thus if this does not hold, we may replace a prefix of P_v by a longer prefix following B. Let u denote the departure point of P_v from B (i.e., u is the last vertex of P_v such that the prefix of P_v from v_0 to u follows edges of B). Let v_k be the knot vertex of B. We split into two cases.

If u lies in P, then we reason as follows. The two paths from u to v_k (in P) respectively v (in P_v) form a chord path for C. Let C' be a reconfiguration of C by this path, and use $P' = P \cap P_v$ as path to attach C' to v_0, defining a balloon $B' = (P', C')$. Decompose B as $P_1 + P_2$, where P_1 and P_2 both end in v, and similarly decompose B' as $P_1' + P_2'$, ending in v. Observe that since u is the new knot vertex, for both possible choices of C' it holds that one of the paths P_i' will be identical to P_v, while the other will be either P_1 or P_2. Since $z(P_v) < z(P_1), z(P_2)$, the new balloon B' represents a constraint with a smaller value than B.

Otherwise, the departure point u lies in $C \setminus P$. The suffix of P_v from u to v forms a chord path; reconfiguring by this chord path leaves two options for the new cycle C', with C' either including v_k or excluding v_k. If C' includes v_k, then we may choose v_k as our new knot vertex. As a result, we may use the same argument as in the previous paragraph, noting that one of the two paths in the decomposition of B' was also present in the decomposition of B, whereas the other path is shorter than both previous paths. Otherwise, finally, the knot vertex will be u, and again one of the two paths in the decomposition of B' is present in the decomposition in B, while the other is the new shortest path to v. ☐

We observe a particular consequence useful for finding min-weight balloons.

Corollary 1. *A minimum-weight balloon constraint can be decomposed as* $z_x(P_u) + z_x(P_v) + (x(u) + x(v))/2$, *where* P_u *and* P_v *are shortest paths and* $uv \in E$.

Proof. Let $B = (P, C)$ be a minimum-weight balloon, and let u be a vertex of B with maximum $z_x(u)$-value. Note that $u \in V(C)$, and we can select u so that u is not the knot vertex. Hence we can decompose B into two paths P_1, P_2, where by Lemma 3 at least one of these paths, say P_1, is a shortest path to u. Let v be the neighbour of u in C that does not lie on P_1. If we can select u so that P_2, truncated to end at P_v, is a shortest path to v, then the result will follow. We show that this indeed holds.

Let U be the set of vertices u of C with maximum $z_x(u)$-value. Note first that the conclusion is trivial if $|U| = 1$, or if there is a vertex $u \in U$ with $x(u) > 0$. In the remaining case, U will be a stretch of vertices in C, which includes the knot vertex only if $U = V(C)$. If every vertex of U has only one shortest path in B, then let u be one of the end points of the stretch U, chosen so that the shortest path to u goes through U. Then $z_x(v) < z_x(u)$, and the shortest path to v in B must be P_2 (using terminology as above). Otherwise, finally, B contains two paths into U of equal length. In this case, we select $u \in U$ arbitrarily, decompose B into $P_1 + P_2$, use P_1 as a shortest path to u, and shorten P_2 into a shortest path of a neighbour v of u, as above. □

2.4 The separation oracle

We now finish the results of this section by constructing a polynomial-time separation oracle over the local LP, assuming a membership oracle for the class $\mathcal{B}$. The result essentially follows from Corollary 1 by slightly perturbing edge lengths so that shortest paths are unique.

Theorem 3. *Let $\Psi = (G = (V, E), \mathcal{B})$ be a biased graph, and let $v_0 \in V$ be the root vertex. Assume that we have access to a polynomial-time membership oracle for $\mathcal{B}$, that for every simple cycle C of G can inform us whether $C \in \mathcal{B}$ or not. Then there is a polynomial-time separation oracle for the local LP rooted in v_0.*

Proof. Let $x : V \to [0, 1]$ be a fractional assignment, where we want to decide whether x is feasible for the LP, i.e., we wish to decide whether there is a v_0-balloon B such that $w(B) < 1$ under x. Order the edges of G as $E = \{e_1, \ldots, e_m\}$. We will associate with each edge $e_i \in E$ a tuple $(\ell_x(e_i), 2^i)$, and modify our distance measure to work componentwise over these tuples, and order distances over such tuples in lexicographical order. This is to deterministically simulate the perturbation of all weights by a random infinitesimal amount. Note that all paths and cycles (in fact, all sets of edges) have unique lengths in this way. In particular, for every vertex u there is a unique shortest path P_u from v_0 to u. Also observe that computations over these weights (and the computations of shortest paths) can be done in polynomial time.

Assume that there is a balloon B with $w(B) < 1$. Then the modified weight of B will be $(w(B), \alpha)$ for some $\alpha \in \mathbb{N}$, where clearly $w(B) < 1$ still holds. Also note that Corollary 1 still applies to the modified weights, since the proof consists only of additions and comparisons. Hence, if B is the (unique) min-weight balloon under the modified weights, then B can be decomposed into two shortest paths P_u, P_v and an edge uv, so that $w(B) = \ell_x(P_u) + \ell_x(P_v) + \ell_x(uv) = z_x(u) + z_x(v) + \ell_x(uv)$. Since shortest paths are unique, we can find the components P_u, P_v and uv defining B, if starting from the edge uv. From these paths, it is easy to reconstruct the cycle C and verify via an oracle query that $C \notin \mathcal{B}$. By iterating the procedure over all edges $uv \in E$, we will find the min-weight balloon. (We also note that for any edge $uv \in E$, the pair of paths P_u and P_v either use the edge uv, so that one of them is a prefix of the other, or form a "balloon shape" with uv, i.e., a shared prefix path followed by a cycle. However, this observation is not required for the proof.) □

3 Half-integrality and persistence of the local LP

We now proceed to use the insights gained in the previous section to show that the local LP is in fact half-integral and obeys a strong persistence property. The approach of the proof is closely related to that of Guillemot for variants of Multiway Cut [12].

3.1 Half-integrality

Let $c : V \to \mathbb{Q}$ be arbitrary vertex weights, let x^* be an optimal solution to the above LP, and let y^* be an optimum solution to the dual. By complementary slackness, if $y^*_B > 0$ then $w(B) = 1$ under x^*, and if $x^*_v > 0$ then the packing y^* saturates v to capacity c_v. Let V_R denote the set of vertices reachable from v_0 at distance 0 (also implying that $x^*_v = 0$ for every $v \in V_R$). We let $V_1 = \{v \in V : x_v = 1\}$ and $V_{1/2} = N(V_R) \setminus V_1$. We claim that $\frac{1}{2}V_{1/2} + V_1$ is a new LP-optimum. This will follow relatively easily from the "path pair perspective" on balloons.

First, we give a structural lemma.

Lemma 4. *Let $B = (P, C)$ be a balloon with $w(B) = 1$ with respect to the vertex weights x^*. Then B is of one of the following types:*

1. *B is contained within $G[V_R + v]$ for some vertex v, where $v \in V_1 \cap (V(C) \setminus V(P))$;*
2. *$B \cap N(V_R)$ is a single vertex $v \in V_{1/2}$, and $v \in V(P)$;*
3. *$B \cap N(V_R)$ consists of two vertices u, v which cut C "in half" as follows: C consists of two uv-paths of which one is contained in V_R and contains at least one internal vertex v' which is the knot vertex of B, and the other path is disjoint from V_R.*

Proof. Since x^* is a feasible solution for the local LP, any balloon with $w(B) = 1$ is a min-weight balloon. Thus by Corollary 1, we can decompose B as $P_u + P_v + uv$ where u and v are shortest paths. In particular, both P_u and P_v contain a prefix in V_R, make at most one visit to $N(V_R)$, and proceed to subsequently not revisit $N[V_R]$ at all. Also note that B necessarily intersects $N(V_R)$, since B is a connected subgraph rooted in v_0 but not contained in V_R.

First assume that B intersects a vertex of $v_1 \in V_1$. Then since $w(B) = 1$, only one of the paths P_u, P_v contains v_1, whereas the other path is entirely contained in V_R. But then we must have $v_1 \in \{u, v\}$, as otherwise the edge uv cannot exist. We conclude that in this case, B is a balloon of type 1.

Next, assume that B intersects $N(V_R)$ in a single vertex $v' \in V_{1/2}$. We claim that B must intersect v' with a coefficient of 2: Indeed, if not, then one of the paths P_u and P_v, say P_u, must be entirely contained in V_R, in which case $v \in N(u)$ must be contained in $N(V_R)$ and we are back in the previous case (contradicting $v' \in V_{1/2}$). Thus $v' \in V(P)$, and B is a balloon of type 2.

Finally, assume that B intersects $N(V_R)$ in at least two vertices. Then in fact B intersects $N(V_R)$ in exactly two vertices u', v', by the properties of P_u and P_v, and neither of these vertices are in V_1, since $w(B) = 1$. Furthermore, since each of P_u and P_v intersects $N(V_R)$ only once, these vertices u', v' lie after the common part of P_u and P_v, i.e., in $V(C) \setminus V(P)$. Then indeed the knot vertex lies in V_R, whereas the path from u' to v' via uv lies outside of V_R, as required. ☐

We now show that complementary slackness implies that the new fractional assignment is actually an LP-optimum.

Lemma 5. *The assignment $V_1 + \frac{1}{2}V_{1/2}$ is an LP-optimum for the local LP.*

Proof. We first show that $V_1 + \frac{1}{2}V_{1/2}$ is a valid LP-solution. Assume towards a contradiction that $w(B) < 1$ for some balloon $B = (P, C)$ under the proposed weights. Note that $V_1 \cup V_{1/2}$ intersects every balloon since $G[V_R]$ is balanced. If $w(B) < 1$ we must thus have $B \cap (V_1 \cup V_{1/2}) = \{v\}$ for some $v \in V_{1/2}$, where $v \in (V(C) \setminus V(P))$. But then $V(B) \subseteq V_R \cup \{v\}$ (since otherwise C passes the "border" $N(V_R)$ in at least two locations), which implies $x^*_v = 1$, contrary to assumptions.

We now show optimality. By complementary slackness, y^* is a packing of balloons saturating every $v \in N(V_R)$ to its capacity c_v, with $w(B) = 1$ for every balloon B in the support of y^*;

hence B will be of one of the three types of Lemma 4. For $i = 1, 2, 3$, let $\mathcal{B}_i$ be the set of balloons of type i from the support of y^*. By optimality, $c^T x^* = \sum_B y^*_B$. Note that a balloon of type 1 intersects one vertex in V_1 with coefficient 1 and no other vertex in $V_1 \cup V_{1/2}$, while a balloon of type 2 or 3 intersects one vertex in $V_{1/2}$ with coefficient 2, respectively two vertices in $V_{1/2}$ with coefficient 1 each, and no other vertex from $V_1 \cup V_{1/2}$. We get

$$\sum_{B \in \mathcal{B}_1} y^*_B = \sum_{v \in V_1} c_v$$

and

$$\sum_{B \in \mathcal{B}_2 \cup \mathcal{B}_3} 2y^*_B = \sum_{v \in V_{1/2}} c_v$$

thus

$$c^T x^* = \sum_B y^*_B = c^T (V_1 + \frac{1}{2} V_{1/2})$$

which shows that $V_1 + \frac{1}{2} V_{1/2}$ is an LP-optimum. □

3.2 Persistence and the tiling property

Finally, we reach the statement concerning the persistence of the local LP. Since the statement is somewhat intricate, let us walk through it. First of all, the basic persistence property is similar to that used in Multiway Cut [11, 12, 6]. Let V_R be defined as before, and for a solution $X \subseteq V$ to Rooted Biased Graph Cleaning (hence $v_0 \notin X$) define S_X as the set of vertices of the connected component of $G - X$ containing v_0. of $G - X$ containing v_0. Then persistence dictates that there is an optimal solution X such that if $z_x(v) = 0$, i.e., if $v \in V_R$ then $v \in S_X$, and if $v \in V_1$ then $v \in X$. We note that both these properties hold for the computed set S' in the below lemma.

However, the lemma also gives a useful tiling property, for the purposes of solving the global problem: Let X be a solution to the full, global Biased Graph Cleaning problem, with $v_0 \notin X$, and let S be the vertices reachable from v_0 in $G - X$. Then it “does not hurt” the global solution to assume that the induced solution $X \cap N(S)$ to the local problem also observes the persistence properties as above. This is implied by the closed-neighbourhood condition on S^+: We will “cut away” a section $G[S^+]$ of the initial graph, and find a new solution for it such that all vertices of S^+ neighbouring $V \setminus S^+$ are deleted, and so that the solution in $G[S^+]$ respects persistence properties while not being more expensive than the original solution $X \cap S^+$. This allows us to assemble a solution to the global problem out of pieces computed for instances of the local problem.

Lemma 6. *Let $x = V_1 + \frac{1}{2} V_{1/2}$ be the half-integral optimum from above, and let V_R be the corresponding reachable region. Let S be a balanced set with $v_0 \in S$. Then we can grow the closed region $N[S]$ to $N[S \cup V_R]$ without paying a larger cost for deleting vertices. More formally, there is a set of vertices S^+ and a set $S' \subseteq S^+$ such that $G[S']$ is balanced and the following hold.*

1. *$S^+ = N[S \cup V_R]$;*
2. *$N[S'] \subseteq S^+$;*
3. *$V_R \subseteq S'$;*
4. *$V_1 \subseteq (S^+ \setminus S')$;*

5. $c(S^+ \setminus S') \leq c(N(S))$.

Proof. Let U be the connected component of v_0 in $G[S \cap V_R]$. We define the sets $S^+ = N[S \cup V_R]$, and

$$S' = V_R \cup (N(U) \cap V_{1/2} \cap S) \cup (S \setminus N[V_R]).$$

Observe that $S' \subseteq S \cup V_R$ and that $V_1 \subseteq N(S')$. We first show that $G[S']$ is balanced. Assume not, and let C be an unbalanced cycle contained in $G[S']$. We may assume that C is reduced as by Lemma 1 with respect to V_R. Observe that C must intersect V_R, as otherwise $V(C) \subseteq S$ contradicting that $G[S]$ is balanced. By Lemma 1, this intersection takes the form of a simple path P_{ab} connecting two vertices $a, b \in N(V_R)$. Furthermore, we have $a, b \in N(U) \cap V_{1/2} \cap S$, and the path P_{ab} intersects $V_R \setminus S$, i.e., P_{ab} contains internal vertices not contained in U. Let P_a respectively P_b be the prefix respectively suffix of P_{ab} contained in U, if any, together with the vertices a respectively b. Let P'_{ab} be a new chord path connecting P_a and P_b in U, and let C' be a new cycle resulting from the reconfiguration of C by P'_{ab}. Since C' cannot be contained in S, this cycle must be formed from $P_{ab} + P'_{ab}$, and since C' cannot be contained in V_R it must still contain the vertices a and b (i.e., we had $P_a = a$ and $P_b = b$). But then $V(C') \cap V_R$ consists of two distinct paths; use a chord path P'' between these paths to reconfigure C' into a new cycle C''. Then we may see that $|C'' \cap N(V_R)| = 1$, namely only of the vertices $\{a, b\}$, contradicting that $a, b \in V_{1/2}$. We conclude that S' is balanced.

Items 1–4 in the lemma hold by definition or are easy, hence it remains to show that $c(S^+ \setminus S') \leq c(N(S))$. Let us break down this expression. First note that $S^+ = V_R \cup S \cup N(V_R) \cup N(S)$, hence $S^+ \setminus S' \subseteq N(V_R) \cup N(S)$ by definition of S'; more carefully,

$$S^+ \setminus S' = (N(S) \setminus S') \cup (N(V_R) \setminus S').$$

Vertices of $N(S) \setminus S'$ contribute equally to both sides of the inequality and can be ignored, hence we are left with vertices of $N(V_R) \setminus (S' \cup N(S))$ contributing to the left hand side and vertices of $N(S) \cap S' = N(S) \cap V_R$ contributing to the right hand side. Splitting $N(V_R) = V_1 \cup V_{1/2}$, the former set simplifies to $(V_1 \setminus N(S)) \cup (V_{1/2} \setminus (N(U) \cup N(S))$. Relaxing slightly we define

$$Z := (V_1 \setminus N(S)) \cup (V_{1/2} \setminus N(U))$$

and

$$Y := N(U) \cap V_R;$$

it will suffice to show $c(Z) \leq c(Y)$. This will occupy the rest of the proof.

Let y^* be the dual optimum, i.e., a fractional packing of balloons which saturates v for every $v \in Z$, with each balloon B in the support being of types 1–3 of Lemma 4 (by complementary slackness). Note that every vertex of Z is in the support of x. Let $\mathcal{B}_1$ contain the balloons from the support of y^* which intersect Z with a total coefficient of 1 (i.e., $\mathcal{B}_1$ contains balloons of type 1, and balloons of type 3 which intersect Z in only one vertex), and let $\mathcal{B}_2$ contain those which intersect Z with a total coefficient of 2 (i.e., balloons of type 2, and balloons of type 3 which intersect Z in two vertices). Note that no balloon from the support of y^* intersects $Z \subseteq N(V_R)$ with a total coefficient of more than 2. Then

$$c(Z) = \sum_{B \in \mathcal{B}_1} y^*_B + \sum_{B \in \mathcal{B}_2} 2y^*_B.$$

We need to show that every $B \in \mathcal{B}_1$ intersects Y with a total coefficient of at least 1, and every $B \in \mathcal{B}_2$ intersects Y with a total coefficient of at least 2. The inequality will follow.

First consider $v \in V_1 \cap Z$, and let $B \in \mathcal{B}_1$ intersect v. Then B is of type 1, hence contained in $G[V_R + v]$. If $B - v \subseteq U$, then $B \cap N(S) = \{v\}$, but $v \in Z$ implies $v \notin N(S)$; hence not all of $B - v$ is contained in U, and B intersects Y.

Next, consider a vertex $v \in V_{1/2} \cap Z$ and a balloon $B \in \mathcal{B}_1$ intersecting v with coefficient 1; hence B is of type 3. Since $v \in Z$ we have $v \notin N(U)$; since B connects $v_0 \in U$ with $v \notin N[U]$ using internal vertices in V_R, there must exist some vertex $u \in B$ contained in $N(U) \cap V_R = Y$. Hence B intersects Y.

Thirdly, consider some $v \in V_{1/2} \cap Z$ intersecting some $B \in \mathcal{B}_2$ with a coefficient of 2. Again $v \notin N(U)$, and B contains a path from v_0 to v with internal vertices in V_R; furthermore every vertex on this path has coefficient 2 in B. Thus B intersects Y with a coefficient of at least 2.

Finally, consider a balloon $B \in \mathcal{B}_2$ of type 3, intersecting two vertices $v, v' \in V_{1/2} \cap Z$. Then B traces two paths from v_0 to v, v', and either both these paths intersect Y in a single vertex (which then has coefficient 2), or two distinct vertices of Y intersect B (for a total coefficient of 2).

This shows that every balloon B in the support of y^* intersects Y with at least as large a total coefficient as it intersects Z. Since y^* is a packing that saturates Z and does not over-saturate any vertex, we have $c(Y) \geq \sum_{\mathcal{B}_1} y^*_B + \sum_{\mathcal{B}_2} 2y^*_B = c(Z)$ as promised. This finishes the proof. □

4 The FPT algorithms

We finally wrap up by giving our main results.

4.1 Rooted Biased Graph Cleaning

We use the results of the previous section to finalise Theorem 1. We proceed by lemmas. Throughout, we assume access to a membership oracle for the biased graph so that we can optimise the LP.

Lemma 7. Rooted Biased Graph Cleaning *admits a 2-approximation, even for weighted graphs.*

Proof. Let $x^* \in [0,1]^V$ be an LP-optimum computed via Theorem 3. Compute the sets V_R, V_1 and $V_{1/2}$ from x^* as in Section 3.1, forming a half-integral optimum x. It is now clear that the set $X = V_1 \cup V_{1/2}$ is an integral solution and a 2-approximation to the problem. □

Lemma 8. Rooted Biased Graph Cleaning *for unweighted graphs can be solved in $O^*(4^{k-\lambda})$ time, where λ is the optimum of the local LP.*

Proof. Assume $k < n$, as otherwise we may simply accept the instance. We will execute a branching process, repeatedly selecting half-integral vertices and recursively "forcing" them to take values $x_v = 0$ or $x_v = 1$.

More precisely, we will use the terms *fix* $v = 0$ and *fix* $v = 1$ for the following procedures. To fix $v = 0$, we simply set $c_v = 2n$; since the LP is half-integral in the presence of vertex weights, this implies that either $x_v = 0$ or the LP optimum takes a total cost of at least $n > k$ and the current branch is terminated. The latter, in turn, can only happen if there is a balloon contained in the set of vertices fixed to 0, as it would otherwise be cheaper to delete every other vertex of G, and in such a case the termination of the current branch of our branching process is correct. To fix $v = 1$, we similarly set $c_v = (1/3n)$. This implies that either $x_v \geq 1/2$ in the LP-solution returned, or we are looking at a branch where some other set of vertices fixed to $v' = 1$ already intersect all balloons passing through v, in which case our current branch cannot produce a minimum solution anyway. Hence, if we ever receive an LP-optimum where $v > 0$ for a vertex v considered fixed to $x_v = 0$, or where $v = 0$ for a vertex v considered fixed to $x_v = 1$, then we may terminate the current branch.

We now describe the branching process. Compute a half-integral optimum $V_1 + \frac{1}{2}V_{1/2}$ for the local LP as above, and reject the instance if $\lambda \geq k + 1/2$. Adjust this optimum such that the corresponding set V_R is as large as possible, by repeatedly fixing $v = 0$ for non-fixed vertices $v \in V_{1/2}$, and keeping the assignment if the cost of the LP does not increase. As observed previously, since this is an LP-optimum of a persistent LP we may fix $x_v = 0$ for every $v \in V_R$ and $x_v = 1$ for every $v \in V_1$ without losing any integral optimum. If this leaves $V_{1/2}$ devoid of unfixed vertices, then we may simply round up any vertices v fixed to $v = 1$ but with $v \in V_{1/2}$ to $x_v = 1$, and we produce a set $X = N(V_R)$ with $|X| = \lambda \leq k$ and we are done. Otherwise, we select an unfixed vertex $v \in V_{1/2}$ and branch recursively on fixing $v = 0$ and fixing $v = 1$, in the latter branch reducing our budget k by one, and repeat the above process to exhaustion. Then in the branch $v = 0$, λ increases by at least $1/2$ since it is the cost of a half-integral LP solution with a value larger than λ, whereas in the branch $v = 1$ λ decreases by $1/2 - (1/3n)$ due to the change of vertex cost, whereas k decreases by 1. Hence in both branches the value of $k - \lambda$ decreases by (very close to) $1/2$, which means that after a branching depth of at most $2(k - \lambda)$, each branch has either terminated or produced a solution. (Any additional contributions of $c/(3n)$ will add up to strictly less than $1/2$, making the process safe.) Thus the total running time is $O^*(2^{2(k-\lambda)})$ as promised. ☐

Lemma 9. Rooted Biased Graph Cleaning *for unweighted graphs can be solved in $O^*(2^k)$ time.*

Proof. If $\lambda \leq k/2$, then we may produce a solution of size at most k by rounding up the LP-optimum. Otherwise $\lambda > k/2$ and the result follows from Lemma 8 since $4^{k-\lambda} = 2^{2(k-\lambda)} < 2^{2(k-k/2)} = 2^k$. ☐

This concludes the proof of Theorem 1.

4.2 Biased Graph Cleaning

We now finally show the full solution to Biased Graph Cleaning.

Proof of Theorem 2. Select an arbitrary vertex $v_0 \in V$ and branch over two options: either delete v_0, or decide that $v_0 \notin X$ and proceed to solve the local problem rooted in v_0. In the latter case, compute a half-integral optimum $V_1 + \frac{1}{2}V_{1/2}$ of the local LP for which the set V_R of reachable vertices is maximal, as in Lemma 8. We claim that under the assumption that there is an optimum $X \subseteq V$ to the global problem with $v_0 \notin X$, there is such an optimum with $V_1 \subseteq X$ and $V_R \subseteq V(H)$ for some connected component H of $G - X$.

For this, let Y be an optimum with $v_0 \notin Y$, and let H be the connected component of $G - Y$ for which $v_0 \in V(H)$. Applying Lemma 6 to $V(H)$ gives us the sets $S^+ = N[V(H) \cup V_R]$ and $S' \supseteq V_R$. Let $Y' = (Y \setminus S^+) \cup (S^+ \setminus S')$. Since $N[V(H)] \subseteq S^+$ we have $N(V(H)) \subseteq Y \cap S^+$, and by Lemma 6 we have $|N(V(H))| \geq |S^+ \setminus S'|$, hence $|Y'| \leq |Y|$. We also have $V_1 \subseteq Y'$, and $G - Y'$ contains a connected component H' with $V_R \subseteq V(H')$. We claim that Y' is a solution. Assume to the contrary that there is some unbalanced cycle C with $V(C) \cap Y' = \emptyset$. Then C intersects Y in $S^+ \setminus Y' = S'$. But since $N(S') \subseteq Y'$ this contradicts that $G[S']$ is balanced. Hence Y' is also an optimal solution, and the claim is shown.

Hence, we may fix $v = 0$ for every $v \in V_R$, and $v = 1$ for every $v \in V_1$, and proceed as in Lemma 8 until the vertices fixed to 0 contain a connected component containing v_0, surrounded entirely by vertices fixed to 1. In such a case, we simply proceed as above with a new starting vertex v_0 in a non-balanced connected component of G, until we either exceed our budget k or discover an integral solution X, and we are done. As in Lemma 8, while branching on a local LP the gap between lower bound and remaining budget decreases in both branches, whereas branching on a new vertex v_0 will certainly increase the solution cost, since the previous solution at this point does not account for any vertices in the connected component of v_0. Hence we have a tree with a branching factor of 2 and a height of at most $2k$, implying a total size and running time of $O^*(4^k)$, and we are done. $\square$

5 Conclusions

We have shown that the combinatorial notion of *biased graphs*, especially the notion of *co-linear cycle classes*, allows us to formulate an LP-branching FPT algorithm for a surprisingly broad class of problems, including the full generality of the Biased Graph Cleaning parameterized by k, and Rooted Biased Graph Cleaning parameterized by relaxation gap. Compared to previous results [14], these algorithms are somewhat more general, and significantly more grounded in combinatorial notions. Open problems include completely combinatorial FPT algorithms, and settling the associated kernelization questions.

References

[1] R. H. Chitnis, M. Cygan, M. Hajiaghayi, M. Pilipczuk, and M. Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. In *FOCS*, pages 460–469, 2012.

[2] F. A. Chudak, M. X. Goemans, D. S. Hochbaum, and D. P. Williamson. A primal-dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs. *Oper. Res. Lett.*, 22(4-5):111–118, 1998.

[3] J. Crampton, G. Gutin, and R. Watrigant. An approach to parameterized resiliency problems using integer linear programming. *CoRR*, abs/1605.08738, 2016.

[4] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[5] M. Cygan, M. Pilipczuk, and M. Pilipczuk. On group feedback vertex set parameterized by the size of the cutset. In *WG*, pages 194–205, 2012.

[6] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3, 2013.

[7] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM J. Discrete Math.*, 27(1):290–309, 2013.

[8] M. DeVos, D. Funk, and I. Pivotto. When does a biased graph come from a group labelling? *Advances in Applied Mathematics*, 61:1–18, 2014.

[9] T. A. Dowling. A class of geometric lattices based on finite groups. *J. Combin. Theory Ser. B*, 14:61–86, 1973.

[10] R. Ganian and S. Ordyniak. The complexity landscape of decompositional parameters for ILP. In *AAAI*, pages 710–716. AAAI Press, 2016.

[11] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.

[12] S. Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61–71, 2011.

[13] A. Huber, A. A. Krokhin, and R. Powell. Skew bisubmodularity and valued CSPs. *SIAM J. Comput.*, 43(3):1064–1084, 2014.

[14] Y. Iwata, M. Wahlström, and Y. Yoshida. Half-integrality, LP-branching and FPT algorithms. *SIAM J. Comput.*, 45(4):1377–1411, 2016.

[15] B. M. P. Jansen and S. Kratsch. A structural approach to kernels for ILPs: Treewidth and total unimodularity. In *ESA*, volume 9294 of *Lecture Notes in Computer Science*, pages 779–791. Springer, 2015.

[16] P. Kolman, M. Koutecký, and H. R. Tiwary. Extension complexity, MSO logic, and treewidth. In *SWAT*, volume 53 of *LIPIcs*, pages 18:1–18:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[17] S. Kratsch and M. Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *FOCS*, pages 450–459, 2012.

[18] D. Lokshtanov. Parameterized integer quadratic programming: Variables and coefficients. *CoRR*, abs/1511.00310, 2015.

[19] D. Lokshtanov, N. S. Narayanaswamy, V. Raman, M. S. Ramanujan, and S. Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, 2014.

[20] D. Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.

[21] S. Mishra, V. Raman, S. Saurabh, S. Sikdar, and C. Subramanian. The complexity of König subgraph problems and above-guarantee vertex cover. *Algorithmica*, 61:857–881, 2011. 10.1007/s00453-010-9412-2.

[22] G. Nemhauser and L. Trotter. Vertex packing: structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.

[23] I. Pivotto. Biased graphs and their matroids I. `http://matroidunion.org/?p=161`, 2013. Accessed: 2016-07-13.

[24] I. Pivotto. Biased graphs and their matroids II. `http://matroidunion.org/?p=279`, 2013. Accessed: 2016-07-13.

[25] I. Pivotto. Biased graphs and their matroids III. `http://matroidunion.org/?p=357`, 2013. Accessed: 2016-07-13.

[26] V. Raman, M. S. Ramanujan, and S. Saurabh. Paths, flowers and vertex cover. In *ESA*, pages 382–393, 2011.

[27] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Algorithms and combinatorics. Springer, 2003.

[28] T. Zaslavsky. Biased graphs. I. Bias, balance, and gains. *J. Comb. Theory, Ser. B*, 47(1):32–52, 1989.

[29] T. Zaslavsky. Biased graphs. II. The three matroids. *J. Comb. Theory, Ser. B*, 51(1):46–72, 1991.