Multi-Budgeted Directed Cuts

Stefan Kratsch
Institut für Informatik, Humboldt-Universität zu Berlin
kratsch@informatik.hu-berlin.de

Shaohua Li
Institute of Informatics, University of Warsaw
Shaohua.Li@mimuw.edu.pl

Dániel Marx
Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
dmarx@cs.bme.hu

Marcin Pilipczuk
Institute of Informatics, University of Warsaw
malcin@mimuw.edu.pl

Magnus Wahlström[1]
Royal Holloway, University of London
Magnus.Wahlstrom@rhul.ac.uk

Abstract

In this paper, we study multi-budgeted variants of the classic minimum cut problem and graph separation problems that turned out to be important in parameterized complexity: Skew Multicut and Directed Feedback Arc Set. In our generalization, we assign colors $1, 2, ..., \ell$ to some edges and give separate budgets $k_1, k_2, ..., k_\ell$ for colors $1, 2, ..., \ell$. For every color $i \in \{1, ..., \ell\}$, let E_i be the set of edges of color i. The solution C for the multi-budgeted variant of a graph separation problem not only needs to satisfy the usual separation requirements (i.e., be a cut, a skew multicut, or a directed feedback arc set, respectively), but also needs to satisfy that $|C \cap E_i| \leq k_i$ for every $i \in \{1, ..., \ell\}$.

Contrary to the classic minimum cut problem, the multi-budgeted variant turns out to be NP-hard even for $\ell = 2$. We propose FPT algorithms parameterized by $k = k_1 + ... + k_\ell$ for all three problems. To this end, we develop a branching procedure for the multi-budgeted minimum cut problem that measures the progress of the algorithm not by reducing k as usual, by but elevating the capacity of some edges and thus increasing the size of maximum source-to-sink flow. Using the fact that a similar strategy is used to enumerate all important separators of a given size, we merge this process with the flow-guided branching and show an FPT bound on the number of (appropriately defined) important multi-budgeted separators. This allows us to extend our algorithm to the Skew Multicut and Directed Feedback Arc Set problems.

Furthermore, we show connections of the multi-budgeted variants with weighted variants of the directed cut problems and the Chain ℓ-SAT problem, whose parameterized complexity remains an open problem. We show that these problems admit a bounded-in-parameter number of "maximally pushed" solutions (in a similar spirit as important separators are maximally pushed), giving somewhat weak evidence towards their tractability.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases important separators, multi-budgeted cuts, Directed Feedback Vertex Set, fixed-parameter tractability, minimum cut

Digital Object Identifier 10.4230/LIPIcs.IPEC.2018.18

[1] Supported by EPSRC grant EP/P007228/1

13th International Symposium on Parameterized and Exact Computation (IPEC 2018).
Editors: Christophe Paul and Michał Pilipczuk; Article No. 18; pp. 18:1–18:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version A full version of the paper is available at https://arxiv.org/abs/1810.06848.

Funding This research is a part of projects that have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreements No 714704 (S. Li and M. Pilipczuk), 280152 (D. Marx), and 725978 (D. Marx).

1 Introduction

Graph separation problems are important topics in both theoretical area and applications. Although the famous minimum cut problem is known to be polynomial-time solvable, many well-known variants are NP-hard, which are intensively studied from the point of view of approximation [1, 2, 11, 13, 14, 18] and, what is more relevant for this work, parameterized complexity.

The notion of important separators, introduced by Marx [22], turned out to be fundamental for a number of graph separation problems such as MULTIWAY CUT [22], DIRECTED FEEDBACK VERTEX SET [4], or ALMOST 2-CNF SAT [27]. Further work, concerning mostly undirected graphs, resulted in a wide range of involved algorithmic techniques: applications of matroid techniques [19, 20], shadow removal [8, 25], randomized contractions [5], LP-guided branching [10, 15, 16, 17], and treewidth reduction [24], among others.

From the above techniques, only the notion of important separators and the related technique of shadow removal generalizes to directed graphs, giving FPT algorithms for DIRECTED FEEDBACK ARC SET [4], DIRECTED MULTIWAY CUT [8], and DIRECTED SUBSET FEEDBACK VERTEX SET [7]. As a result, the parameterized complexity of a number of important graph separation problems in directed graphs remains open, and the quest to investigate them has been put on by the third author in a survey from 2012 [23]. Since the publication of this survey, two negative answers have been obtained. Two authors of this work showed that DIRECTED MULTICUT is W[1]-hard even for four terminal pairs (leaving the case of three terminal pairs open) [26], while Lokshtanov et al. [21] showed intractability of DIRECTED ODD CYCLE TRANSVERSAL.

During an open problem session at Recent Advancements in Parameterized Complexity school (December 2017) [12], Saurabh posed the question of parameterized complexity of a weighted variant of DIRECTED FEEDBACK ARC SET, where given a directed edge-weighted graph G, an integer k, and a target weight w, the goal is to find a set $X \subseteq E(G)$ such that $G - X$ is acyclic and X is of cardinality at most k and weight at most w. Consider a similar problem WEIGHTED *st*-CUT: given a directed graph G with positive edge weights and two distinguished vertices $s, t \in V(G)$, an integer k, and a target weight w, decide if G admits an *st*-cut of cardinality at most k and weight at most w. The parameterized complexity of this problem parameterized by k is open even if G is restricted to be acyclic, while with this restriction the problem can easily be reduced to DIRECTED FEEDBACK ARC SET (add an arc (t, s) of prohibitively large weight).

The WEIGHTED *st*-CUT problem becomes similar to another directed graph cut problem, identified in [6], namely CHAIN ℓ-SAT. While this problem is originally formulated in CSP language, the graph formulation is as follows: given a directed graph G with a partition of edge set $E(G) = P_1 \uplus P_2 \uplus \ldots \uplus P_m$ such that each P_i is an edge set of a simple path of length at most ℓ (the input paths could have common nodes), an integer k, and two vertices $s, t \in V(G)$, find an *st*-cut $C \subseteq E(G)$ such that $|\{i | C \cap P_i \neq \emptyset\}| \leq k$. This problem can easily be seen to be equivalent to minimum *st*-cut problem (and thus polynomial-time solvable) for $\ell \leq 2$, but is NP-hard for $\ell \geq 3$ and its parameterized complexity (with k as a parameter) remains an open problem.

In this paper we make progress towards resolving the question of parameterized complexity of the two aforementioned problems: weighted st-cut problem (in general digraphs, not necessary acyclic ones) and Chain ℓ-SAT. Our contribution is twofold.

Multi-budgeted variant

We define a *multi-budgeted* variant of a number of cut problems (including the minimum cut problem) and show its fixed-parameter tractability. In this variant, the edges of the graph are colored with ℓ colors, and the input specifies separate budgets for each color. More formally, we primarily consider the following problem.

> Multi-budgeted cut
> **Input:** A directed graph G, two disjoint sets of vertices $X, Y \subseteq V(G)$, an integer ℓ, and for every $i \in \{1, 2, \ldots, \ell\}$ a set $E_i \subseteq E(G)$ and an integer $k_i \geq 1$.
> **Question:** Is there a set of arcs $C \subseteq \bigcup_{i=1}^{\ell} E_i$ such that there is no directed $X - Y$ path in $G \setminus C$ and for every $i \in [\ell]$, $|C \cap E_i| \leq k_i$.

Similarly we can define multi-budgeted variants of Directed Feedback Arc Set and Skew Multicut.

We observe that Multi-budgeted cut for $\ell = 2$ reduces to Weighted st-cut as follows. Let $(G, X, Y, E_1, E_2, k_1, k_2)$ be a Multi-budgeted cut instance for $\ell = 2$. First, observe that we may assume that $E_1 \cap E_2 = \emptyset$, as we can replace every edge $e \in E_1 \cap E_2$ with two copies $e_1 \in E_1 \setminus E_2$ and $e_2 \in E_2 \setminus E_1$. Second, construct an equivalent Weighted st-cut instance (G', s, t, k, w) as follows. To construct G', first add two vertices s, t to G and edges $\{(s, x) | x \in X\}$ and $\{(y, t) | y \in Y\}$ of prohibitively large weight. Assign also prohibitively large weight to every edge $e \in E(G) \setminus (E_1 \cup E_2)$. Assign weight $(k_1 + 1)k_2 + 1$ to every edge $e \in E_1$. For every edge $e \in E_2$, add $k_1 + 1$ copies of e to G' of weight 1 each. Finally, set $k := (k_1 + 1) \cdot k_2 + k_1$ as the cardinality bound and $w := k_1((k_1 + 1)k_2 + 1) + (k_1 + 1)k_2$ as the target weight. The equivalence of the instances follows from the fact that the cardinality bound allows to pick in the solution at most k_2 bundles of $k_1 + 1$ copies of an edge of E_2, while the weight bound allows to pick only k_1 edges of E_1.

Thus, Multi-budgeted cut for $\ell = 2$ corresponds to the case of Weighted st-cut where the weights are integral and both target cardinality and weight are bounded in parameter.[2] This connection was our primary motivation to study the multi-budgeted variants of the cut problems.

Contrary to the classic minimum cut problem, we note that Multi-budgeted Cut becomes NP-hard for $\ell \geq 2$ by a simple reduction from constrained minimum vertex cover problem on bipartite graphs [3].[3] We show that Multi-budgeted Cut is FPT when parameterized by $k = k_1 + \ldots + k_\ell$. For this problem, our branching strategy is as follows. First, note that in the problem definition we assume that each k_i is positive, and thus $\ell \leq k$. A standard application of the Ford-Fulkerson algorithm gives a minimum XY-cut C of size λ and λ edge-disjoint $X - Y$ paths $P_1, P_2, \ldots, P_\lambda$. If C is a solution, then we are done. Similarly, if $\lambda > k$, then there is no solution. Otherwise, we branch which colors of the sought

[2] For a reduction in the other direction, replace every arc e of weight $\omega(e)$ with one copy of color 1 and $\omega(e)$ copies of color 2, and set budgets $k_1 = k$ and $k_2 = w$.

[3] We believe this problem must have been formulated already before and proven to be NP-hard. However, we were not able to find it in the literature. Our own reduction will be available in the full version of the paper.

solution should appear on each paths P_j; that is, for every $i \in [\ell]$ and $j \in [\lambda]$, we guess if $P_j \cap E_i$ contains an edge of the sought solution, and in each guess assign infinite capacities to the edges of wrong color. If this change increased the size of a maximum flow from X to Y, then we can charge the branching step to this increase, as the size of the flow cannot exceed k. The critical insight is that if the size of the minimum flow does not increase (i.e., $P_1, \ldots, P_\lambda$ remains a maximum flow), then a corresponding minimum cut is necessarily a solution. As a result, we obtain the following.

▶ **Theorem 1.** Multi-budgeted Cut *admits an FPT algorithm with running time bound* $\mathcal{O}(2^{k^2\ell} \cdot k \cdot (|V(G)| + |E(G)|))$ *where* $k = \sum_{i=1}^{\ell} k_i$.

The charging of the branching step to a flow increase appears also in the classic argument for bound of the number of important separators [4] (see also [9, Chapter 8]). We observe that our branching algorithm can be merged with this procedure, yielding a bound (as a function of k) and enumeration procedure of naturally defined multi-budgeted important separators. This in turn allows us to generalize our FPT algorithm to Multi-budgeted Skew Multicut and Multi-budgeted Directed Feedback Arc Set.

▶ **Theorem 2.** Multi-budgeted Skew Multicut *and* Multi-budgeted Directed Feedback Arc Set *admit FPT algorithms with running time bound* $2^{\mathcal{O}(k^3 \log k)}(|V(G)| + |E(G)|)$ *where* $k = \sum_{i=1}^{\ell} k_i$.

Bound on the number of pushed solutions

While we are not able to establish fixed-parameter tractability of the weighted variant of the minimum cut problem (even in acyclic graphs) nor of Chain ℓ-SAT, we show the following graph-theoretic statement. Consider a directed graph G with two distinguished vertices $s, t \in V(G)$. For two (inclusion-wise) minimal st-cuts C_1, C_2 we say that C_1 *is closer to* t than C_2 if every vertex reachable from s in $G - C_2$ is also reachable from s in $G - C_1$. A classic submodularity argument implies that there is exactly one closest to t minimum st-cut, while the essence of the notion of important separators is the observation that there is bounded-in-k number of minimal separators of cardinality at most k that are closest to t. In Section 5 we show a similar existential statement for the two discussed problems.

▶ **Theorem 3.** *For every integer k there exists an integer g such that the following holds. Let G be a directed graph with positive edge weights and two distinguished vertices $s, t \in V(G)$. Let $\mathcal{F}$ be a family of all st-cuts that are of minimum weight among all (inclusion-wise) minimal st-cuts of cardinality at most k. Let $\mathcal{G} \subseteq \mathcal{F}$ be the family of those cuts C such that no other cut of $\mathcal{F}$ is closer to t. Then $|\mathcal{G}| \leq g$.*

▶ **Theorem 4.** *For every integers k, ℓ there exists an integer g' such that the following holds. Let $I := (G, s, t, (P_i)_{i=1}^m, k)$ be a* Chain ℓ-SAT *instance that is a yes-instance but $(G, s, t, (P_i)_{i=1}^m, k-1)$ is a no-instance. Let $\mathcal{F}$ be a family of all (inclusion-wise) minimal solutions to I and let $\mathcal{G} \subseteq \mathcal{F}$ be the family of those cuts C such that no other cut of $\mathcal{F}$ is closer to t. Then $|\mathcal{G}| \leq g'$.*

Unfortunately, our proof is purely existential, and does not yield an enumeration procedure of the "closest to t" solutions.

Organization

In this extended abstract, we prove Theorem 1 in Section 3, present the multi-budgeted extension of the notion of important separators (needed for Theorem 2) in Section 4, and sketch the proofs of Theorems 3 and 4 in Section 5.

2 Preliminaries

For an integer n, we denote $[n] = \{1, 2, \ldots, n\}$. For a directed graph G, we use $V(G)$ to represent the set of vertices of G and $E(G)$ to represent the set of directed edges of G. In all multi-budgeted problems, the directed graph G comes with sets $E_i \subseteq E(G)$ for $i \in [\ell]$ which we refer as *colors*. That is, an edge e is *of color* i if $e \in E_i$, and of *no color* if $e \in E(G) \setminus \bigcup_{i=1}^{\ell} E_i$. Note that an edge may have many colors, as we do not insist on the sets E_i being pairwise disjoint.

Let X and Y be two disjoint vertex sets in a directed graph G, an *XY-cut* of G is a set of edges C such that every directed path from a vertex in X to a vertex in Y contains an edge of C. A cut C is *minimal* if no proper subset of C is an XY-cut, and *minimum* if C is of minimum possible cardinality. Let C be an XY-cut and let R be the set of vertices reachable from X in $G \setminus C$. We define $\delta^+(R) = \{(u, v) \in E(G) | u \in R \text{ and } v \notin R\}$ and note that if C is minimal, then $\delta^+(R) = C$.

Let $(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ be a MULTI-BUDGETED CUT instance and let C be an XY-cut. We say that C is *budget-respecting* if $C \subseteq \bigcup_{i=1}^{\ell} E_i$ and $|C \cap E_i| \leq k_i$ for every $i \in [\ell]$. For a set $Z \subseteq E(G)$ we say that C is *Z-respecting* if $C \subseteq Z$. In such contexts, we often call Z the *set of deletable edges*. An XY-cut C is a *minimum Z-respecting cut* if it is a Z-respecting XY-cut of minimum possible cardinality among all Z-respecting XY-cuts.

Our FPT algorithms start with $Z = \bigcup_{i=1}^{\ell} E_i$ and in branching steps shrink the set Z to reduce the search space. We encapsulate our use of the classic Ford-Fulkerson algorithm in the following statement.

▶ **Theorem 5.** *Given a directed graph G, two disjoint sets $X, Y \subseteq V(G)$, a set $Z \subseteq E(G)$, and an integer k, one can in $\mathcal{O}(k(|V(G)| + |E(G)|))$ time either find the following objects:*

- *λ paths $P_1, P_2, \ldots, P_\lambda$ such that every P_i starts in X and ends in Y, and every edge $e \in Z$ appears on at most one path P_i;*
- *a set $B \subseteq Z$ consisting of all edges of G that participate in some minimum Z-respecting XY-cut;*
- *a minimum Z-respecting XY-cut C of size λ that is closest to Y among all minimum Z-respecting XY-cuts;*

or correctly conclude that there is no Z-respecting XY-cut of cardinality at most k.

3 Multi-budgeted cut

We now give an FPT algorithm parameterized by $k = \Sigma_{i=1}^{\ell} k_i$ for the MULTI-BUDGETED CUT problem. We follow a branching strategy that recursively reduces a set Z of deletable edges. That is, we start with $Z = \bigcup_{i=1}^{\ell} E_i$ (so that every solution is initially Z-respecting) and in each recursive step, we look for a Z-respecting solution and reduce the set Z in a branching step.

Consider a recursive call where we look for a Z-respecting solution to the input MULTI-BUDGETED CUT instance $(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$. That is, we look for a Z-respecting budget-respecting cut. We apply Theorem 5 to it. If it returns that there is no Z-respecting XY-cut of size at most k, we terminate the current branch, as there is no solution. Otherwise, we obtain the paths $P_1, P_2, \ldots, P_\lambda$, the set B (which we will not use in this section), and the cut C.

If C is budget-respecting, then it is a solution and we can return it. Otherwise, we perform the following branching step. We iterate over all tuples $(A_1, ..., A_\ell)$ such that for every $i \in [\ell]$, $A_i \subseteq [\lambda]$ and $|A_i| \leq k_i$. A_i represents the subset of paths $P_1, ..., P_\lambda$ on which at

MultiBudgetedCut$(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$
Input: A directed graph G, two disjoint set of vertices $X, Y \subseteq V(G)$, an integer ℓ, for every $i \in [\ell]$ a set $E_i \subseteq E(G)$ and an integer k.
Output: an XY cut $C \subseteq \bigcup_{i=1}^{\ell} E_i$ such that for every $i \in [\ell]$, $|C \cap E_i| \leq k_i$ if it exists, otherwise return NO.

1. $Z := \bigcup_{i=1}^{\ell} E_i$;
2. **return Solve**(Z);

Solve(Z)
a. apply Theorem 5 to (G, X, Y, k, Z) where $k = \sum_{i=1}^{\ell} k_i$, obtaining objects $(P_i)_{i=1}^{\lambda}$, B, and C, or an answer NO;
b. **if** the answer NO is obtained, **then return** NO;
c. **if** C is budget-respecting, **then return** C;
d. **for each** $(A_1, ..., A_\ell)$ such that for every i in $[\ell]$, $A_i \subseteq [\lambda]$ and $|A_i| \leq k_i$ **do**
d.1 $\widehat{Z} := Z$;
d.2 **for each** $i \in [\ell]$ **do**
  for each $j \in [\lambda] \setminus A_i$ **do**
    $\widehat{Z} := \widehat{Z} \setminus (E_i \cap E(P_j))$;
d.3 $D = $ **Solve**$(\widehat{Z})$;
d.4 **if** $D \neq$NO **then return** D;
e. **return** NO;

Figure 1 FPT algorithm for MULTI-BUDGETED CUT.

least one edge of color i is in the solution for each $i \in [\ell]$. For those edges of color i which are on the paths not indicated by A_i, they are not in the solution. Thus we can safely delete them from Z. More formally, for every $i \in [\ell]$ and $j \in [\lambda] \setminus A_i$, we remove from Z all edges of $E(P_j) \cap E_i$. We recurse on the reduced set Z. A pseudocode is available in Figure 1.

▶ **Theorem 6.** *The algorithm in Figure 1 for* MULTI-BUDGETED CUT *is correct and runs in time* $O(2^{\ell k^2} \cdot k \cdot (|V(G)| + |E(G)|))$ *where* $k = \Sigma_{i=1}^{\ell} k_i$.

Proof. We prove the correctness of the algorithm by showing that it returns a solution if and only if the input instance is a yes-instance. The "only if" direction is obvious, as the algorithm returns only Z-respecting budget-respecting XY-cuts and $Z \subseteq \bigcup_{i=1}^{\ell} E_i$ in each recursive call.

We prove the correctness for the "if" direction. Let C_0 be a solution, that is, a budget-respecting XY-cut. In the initial call to **Solve**, C_0 is Z-respecting. It suffices to inductively show that in each call to **Solve** such that C_0 is Z-respecting, either the call returns a solution, or C_0 is $\widehat{Z}$-respecting for at least one of the subcalls. Since C_0 is Z-respecting, the application of Theorem 5 returns objects $(P_i)_{i=1}^{\lambda}$, B, and C. If C is budget-respecting, then the algorithm returns it and we are done. Otherwise, consider the branch $(A_1, A_2, \ldots, A_\ell)$ where $A_i = \{j | E(P_j) \cap C_0 \neq \emptyset\}$. Since C_0 is budget-respecting, $C_0 \subseteq Z$, and no edge of Z appears on more than one path P_j, we have $|A_i| \leq k_i$ for every $i \in [\ell]$. Thus, $(A_1, A_2, \ldots, A_\ell)$ is a branch considered by the algorithm. In this branch, the algorithm refines the set Z to $\widehat{Z}$. By the definition of A_i, for every $i \in [\ell]$ and $j \in [\lambda] \setminus A_i$, we have $C_0 \cap E_i \cap E(P_j) = \emptyset$. Consequently, C_0 is $\widehat{Z}$-respecting and we are done.

For the time bound, the following observation is crucial.

▶ **Claim 7.** *Consider one recursive call* ***Solve***(Z) *where the application of Theorem 5 in line (a) returned objects* $(P_i)_{i=1}^{\lambda}$, B *and* C. *Assume that in some recursive subcall* ***Solve***$(\widehat{Z})$ *invoked in line (d.3) (Figure 1), the subsequent application of Theorem 5 in line (a) of the* *subcall returned a cut of the same size, that is, the algorithm of Theorem 5 returned a cut $\widehat{C}$ of size $\widehat{\lambda} = \lambda$. Then the cut $\widehat{C}$ is budget-respecting and, consequently, is returned in line (c) of the subcall.*

Proof. Since $|\widehat{C}| = \lambda$ is a $\widehat{Z}$-respecting XY-cut, $\widehat{Z} \subseteq Z$, and every edge $e \in Z$ appears on at most one path P_i, we have that $\widehat{C}$ consists of exactly one edge of $\widehat{Z}$ on every path P_i, that is, $\widehat{C} = \{e_1, e_2, \ldots, e_\lambda\}$ and $e_j \in E(P_j) \cap \widehat{Z}$ for every $j \in [\lambda]$. In other words, the paths $(P_j)_{j=1}^{\lambda}$ still correspond to a maximum flow from X to Y with edges of $\widehat{Z}$ being of unit capacity and edges outside $\widehat{Z}$ of infinite capacity because $(P_j)_{j=1}^{\lambda}$ are paths satisfying that any two of them are disjoint on $\widehat{Z} \subseteq Z$ and λ is still equal to the size of the maximum flow. If $e_j \in E_i$ for some $j \in [\lambda]$ and $i \in [\ell]$, then by the construction of set $\widehat{Z}$, we have $j \in A_i$. Consequently, $|\{j | e_j \in E_i\}| \leq |A_i| \leq k_i$ for every $i \in [\ell]$, and thus $\widehat{C}$ is budget-respecting. ⌟

Claim 7 implies that the depth of the search tree is bounded by k, as the algorithm terminates when λ exceeds k. At every step, there are at most $(2^\lambda)^\ell \leq (2^k)^\ell$ different tuples $(A_1, ..., A_\ell)$ to consider. Consequently, there are $\mathcal{O}(2^{(k-1)k\ell})$ nodes of the search tree that enter the loop in line (d) and $\mathcal{O}(2^{k^2\ell})$ nodes that invoke the algorithm of Theorem 5. As a result, the running time of the algorithm is $\mathcal{O}(2^{\ell k^2} \cdot k \cdot (|V(G)| + |E(G)|))$. ◀

4 Multi-budgeted important separators with applications

Similar to the concept of important separators proposed by Marx [22] (see also [9, Chapter 8]), we define *multi-budgeted important separators* as follows.

▶ **Definition 8.** Let $(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ be a Multi-budgeted cut instance and let $Z \subseteq \bigcup_{i=1}^{\ell} E_i$ be a set of deletable edges. Let C_1, C_2 be two minimal Z-respecting budget-respecting XY-cuts. We say that C_1 *dominates* C_2 if

1. every vertex reachable from X in $G - C_2$ is also reachable from X in $G - C_1$;
2. for every $i \in [\ell]$, $|C_1 \cap E_i| \leq |C_2 \cap E_i|$.

We say that $\widehat{C}$ is an *important Z-respecting budget-respecting XY-cut* if $\widehat{C}$ is a minimal Z-respecting budget-respecting XY-cut and no other minimal Z-respecting budget-respecting XY-cut dominates $\widehat{C}$. $\widehat{C}$ is an *important budget-respecting XY-cut* if it is an *important Z-respecting budget-respecting XY-cut* for $Z = \bigcup_{i=1}^{\ell} E_i$.

Chen et al. [4] showed an enumeration procedure for (classic) important separators using similar charging scheme as the one of the previous section. Our main result in this section is a merge of the arguments from the previous section with the arguments of Chen et al. Theorem 2 follows from Theorem 9 via an analogous arguments as in [4].

▶ **Theorem 9.** *Let $(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ be a* Multi-budgeted cut *instance, let $Z \subseteq \bigcup_{i=1}^{\ell} E_i$ be a set of deletable edges, and denote $k = \sum_{i=1}^{\ell} k_i$. Then one can in $2^{\mathcal{O}(k^2 \log k)}(|V(G)|+|E(G)|)$ time enumerate a family of minimal Z-respecting budget-respecting XY-cuts of size $2^{\mathcal{O}(k^2 \log k)}$ that contains all important ones.*

Proof. Consider the recursive algorithm presented in Figure 2. The recursive procedure **ImportantCut** takes as an input a Multi-budgeted Cut instance $I = (G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ and a set $Z \subseteq \bigcup_{i=1}^{\ell} E_i$, with the goal to enumerate all important Z-respecting budget-respecting XY-cuts. Note that the procedure may output some more Z-respecting budget-respecting XY-cuts; we need only to ensure that

1. it outputs all important ones,
2. it outputs $2^{\mathcal{O}(k^2 \ell \log k)}$ cuts, and
3. it runs within the desired time.

The procedure first invokes the algorithm of Theorem 5 on (G, X, Y, k, Z), where $k = \sum_{i=1}^{\ell} k_i$. If the call returned that there is no Z-respecting XY-cut of size at most k, we can return an empty set. Otherwise, let $(P_j)_{j=1}^{\lambda}$, B, and C be the computed objects. We perform a branching step, with each branch labeled with a tuple $(A_1, A_2, \ldots, A_\ell)$ where $A_i \subseteq [\lambda]$ and $|A_i| \leq k_i$ for every $i \in [\ell]$. A branch $(A_1, A_2, \ldots, A_\ell)$ is supposed to capture important cuts C_0 with $\{j | C_0 \cap B \cap E(P_j) \cap E_i \neq \emptyset\} \subseteq A_i$ for every $i \in [\ell]$; that is, for every $i \in [\ell]$ and $j \in [\lambda]$ we guess if C_0 contains a *bottleneck* edge of color i on path P_j. All this information (i.e., paths P_j, the set B, the cut C, and the sets A_i) are passed to an auxiliary procedure **Enum**.

The procedure **Enum** shrinks the set Z according to sets A_i. More formally, for every $i \in [\ell]$ and $j \in [\lambda] \setminus A_i$ we delete from Z all edges from $B \cap E_i \cap E(P_j)$, obtaining a set $\widehat{Z} \subseteq Z$. At this point, we check if the reduction of the set Z to $\widehat{Z}$ increased the size of minimum Z-respecting XY-cut by invoking Theorem 5 on $(G, X, Y, k, \widehat{Z})$ and obtaining objects $(\widehat{P}_j)_{j=1}^{\widehat{\lambda}}, \widehat{B}, \widehat{C}$ or a negative answer. If the size of the minimum cut increased, that is, $\widehat{\lambda} > \lambda$, we recurse with the original procedure **ImportantCut**. Otherwise, we add one cut to $\mathcal{S}$, namely $\widehat{C}$. Furthermore, we try to shrink one of the sets A_i by one and recurse; that is, for every $i \in [\ell]$ and every $j \in A_i$, we recurse with the procedure **Enum** on sets $A'_{i'}$ where $A'_i = A_i \setminus \{j\}$ and $A'_{i'} = A_{i'}$ for every $i' \in [\ell] \setminus \{i\}$.

Let us first analyze the size of the search tree. A call to **ImportantCut** invokes at most $\binom{\lambda\ell}{\leq k} \leq (k\ell+1)^k$ calls to **Enum**. Each call to **Enum** either falls back to **ImportantCut** if $\widehat{\lambda} > \lambda$ or branches into $\sum_{i=1}^{\ell} |A_i| \leq k\ell$ recursive calls to itself. In each recursive call, the sum $\sum_{i=1}^{\ell} |A_i|$ decreases by one. Consequently, the initial call to **Enum** results in at most $(k\ell)^k$ recursive calls, each potentially falling back to **ImportantCut**. Since each recursive call to **ImportantCut** uses strictly larger value of λ, which cannot grow larger than k, and $\ell \leq k$, the total size of the recursion tree is $2^{\mathcal{O}(k^2 \log k)}$. Each recursive call to **Enum** adds at most one set to $\mathcal{S}$, while each recursive call to **ImportantCut** and **Enum** runs in time $\mathcal{O}(2^{k\ell} \cdot k \cdot (|V(G)| + |E(G)|))$. The promised size of the family $\mathcal{S}$ and the running time bound follows. It remains to show correctness, that is, that every important Z-respecting budget-respecting XY-cut is contained in $\mathcal{S}$ returned by a call to **ImportantCut**(I, Z).

We prove by induction on the size of the recursion tree that (1) every call to **ImportantCut**(I, Z) enumerates all important Z-respecting budget-respecting XY-cuts, and (2) every call to **Enum**$(I, Z, (P_j)_{j=1}^{\lambda}, B, C, (A_i)_{i=1}^{\ell})$ enumerates all important Z-respecting budget-respecting XY-cuts C_0 with the property that $\{j | E_i \cap E(P_j) \cap B \cap C_0 \neq \emptyset\} \subseteq A_i$ for every $i \in [\ell]$.

The inductive step for a call **ImportantCut**(I, Z) is straightforward. Let us fix an arbitrary important Z-respecting budget-respecting XY-cut C_0. Since C_0 is budget-respecting, C_0 is a Z-respecting cut of size at most k, and thus the initial call to Theorem 5 cannot return NO. Consider the tuple $(A_1, A_2, \ldots, A_\ell)$ where for every $i \in [\ell]$, $\{j | E(P_j) \cap E_i \cap B \cap C_0\} = A_i$. Since C_0 is budget-respecting and the paths P_j do not share an edge of Z, we have that $|A_i| \leq k_i$ for every $i \in [\ell]$ and the algorithm considers this tuple in one of the branches. Then, from the inductive hypothesis, the corresponding call to **Enum** returns a set containing C_0.

Consider now a call to **Enum**$(I, Z, (P_j)_{j=1}^{\lambda}, B, C, (A_i)_{i=1}^{\ell})$ and an important Z-respecting budget-respecting XY-cuts C_0 with the property that $\{j | E_i \cap E(P_j) \cap B \cap C_0 \neq \emptyset\} \subseteq A_i$ for every $i \in [\ell]$. By the construction of $\widehat{Z}$ and the above assumption, C_0 is $\widehat{Z}$-respecting. In particular, the call to the algorithm of Theorem 5 cannot return NO. Hence, in the case when $\widehat{\lambda} > \lambda$, C_0 is enumerated by the recursive call to **ImportantCut** and we are done. Assume then $\widehat{\lambda} = \lambda$.

ImportantCut(I, Z)
Input: A MULTI-BUDGETED CUT instance $I = (G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ and a set $Z \subseteq \bigcup_{i=1}^{\ell} E_i$.
Output: a family $\mathcal{S}$ of minimal Z-respecting budget-respecting XY-cuts that contains all important ones.
1. $\mathcal{S} := \emptyset$;
2. apply the algorithm of Theorem 5 to (G, X, Y, k, Z) with $k = \sum_{i=1}^{\ell} k_i$, obtaining either objects $(P_i)_{i=1}^{\lambda}$, B, and C, or an answer NO;
3. **if** an answer NO is obtained, **then return** $\mathcal{S}$;
4. **for each** $(A_1, ..., A_\ell)$ such that for every i in $[\ell]$, $A_i \subseteq [\lambda]$ and $|A_i| \leq k_i$ **do**
4.1 $\mathcal{S} := \mathcal{S} \cup$ **Enum**$(I, Z, (P_j)_{j=1}^{\lambda}, B, C, (A_i)_{i=1}^{\ell})$
5. **return** $\mathcal{S}$

Enum$(I, Z, (P_j)_{j=1}^{\lambda}, B, C, (A_i)_{i=1}^{\ell})$
Input: A MULTI-BUDGETED CUT instance $I = (G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$, a set $Z \subseteq \bigcup_{i=1}^{\ell} E_i$, a family $(P_j)_{j=1}^{\lambda}$ of paths from X to Y such that every edge of Z appears on at most one path P_j, a set B consisting of all edges that participate in some minimum Z-respecting XY-cut, a minimum Z-respecting XY-cut C closest to Y, and sets $A_i \subseteq [\lambda]$ of size at most k_i for every $i \in [\ell]$
Output: a family $\mathcal{S}$ of minimal Z-respecting budget-respecting XY-cuts that contains all cuts C_0 that are important Z-respecting budget respecting XY-cuts and satisfy $\{j | E(P_j) \cap B \cap C_0 \cap E_i \neq \emptyset\} \subseteq A_i$ for every $i \in [\ell]$.
a. $\widehat{Z} := Z$;
b. **for each** $i \in [\ell]$ **do**
    for each $j \in [\lambda] \setminus A_i$ **do**
        $\widehat{Z} := \widehat{Z} \setminus (B \cap E_i \cap E(P_j))$;
c. apply the algorithm of Theorem 5 to $(G, X, Y, k, \widehat{Z})$, obtaining either objects $(\widehat{P}_i)_{i=1}^{\widehat{\lambda}}$, $\widehat{B}$, and $\widehat{C}$ or an answer NO;
d. **if** $\widehat{\lambda}$ exists and $\widehat{\lambda} > \lambda$, **then**
d.1 $\mathcal{S} := \mathcal{S} \cup$ **ImportantCut**$(I, \widehat{Z})$;
e. **else if** $\widehat{\lambda}$ exists and equals λ, **then**
e.1 $\mathcal{S} := \mathcal{S} \cup \{\widehat{C}\}$;
e.2 **for each** $i \in [\ell]$ **do**
    for each $j \in A_i$ **do**
        $A'_i := A_i \setminus \{j\}$ and $A'_{i'} := A_{i'}$ for every $i' \in [\ell] \setminus \{i\}$
        $\mathcal{S} := \mathcal{S} \cup$ **Enum**$(I, \widehat{Z}, (P_j)_{j=1}^{\lambda}, \widehat{B}, \widehat{C}, (A'_i)_{i=1}^{\ell})$.
f. **return** $\mathcal{S}$

Figure 2 FPT algorithm for enumerating important multi-budgeted Z-respecting XY-cuts.

For $i \in [\ell]$, let $\widehat{A}_i = \{j | E_i \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset\}$. Since $\widehat{Z} \subseteq Z$ but the sizes of minimum Z-respecting and $\widehat{Z}$-respecting XY-cuts are the same, we have $\widehat{B} \subseteq B$. Consequently, $\widehat{A}_i \subseteq A_i$ for every $i \in [\ell]$.

Assume there exists $i \in [\ell]$ such that $\widehat{A}_i \subsetneq A_i$ and let $j \in A_i \setminus \widehat{A}_i$. Consider then the branch (i, j) of the **Enum** procedure, that is, the recursive call with $A'_i = A_i \setminus \{j\}$ and $A'_{i'} = A_{i'}$ for $i' \in [\ell] \setminus \{i\}$. Observe that we have $\{j | E_{i'} \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset\} \subseteq A'_{i'}$ for every $i' \in [\ell]$ and, by the inductive hypothesis, the corresponding call to **Enum** enumerates C_0. Hence, we are left only with the case $\widehat{A}_i = A_i$, that is, $A_i = \{j | E_i \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset\}$ for every $i \in [\ell]$.

We claim that in this case $C_0 = \widehat{C}$. Assume otherwise. Since $|\widehat{C}| = \widehat{\lambda} = \lambda$ and $\widehat{Z} \subseteq Z$, $\widehat{C}$ contains exactly one edge on every path P_j. Also, $\widehat{C} \subseteq \widehat{B}$ by the definition of the set $\widehat{B}$. Since $\widehat{C}$ is the minimum $\widehat{Z}$-respecting XY-cut that is closest to Y, $\widehat{C} = \{e_1, e_2, \ldots, e_\lambda\}$ where e_j is the last (closest to Y) edge of $\widehat{B}$ on the path P_j for every $j \in [\lambda]$.

Let R_0 and $\widehat{R}$ be the set of vertices reachable from X in $G - C_0$ and $G - \widehat{C}$, respectively.

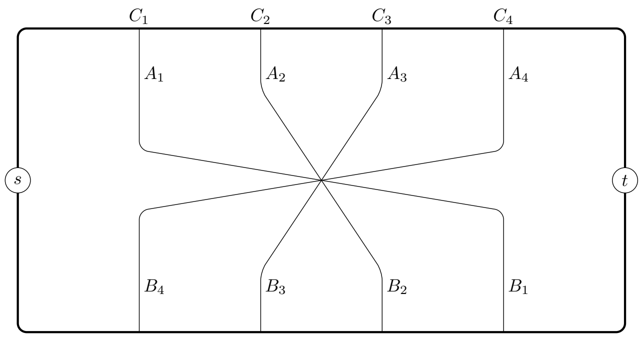

Figure 3 A schematic picture of a 4-bowtie.

Let D be a minimal XY-cut contained in $\delta^+(R_0 \cup \widehat{R})$. (Note that $\delta^+(R_0 \cup \widehat{R})$ is an XY-cut because $X \subseteq R_0 \cup \widehat{R}$ and $Y \cap (R_0 \cup \widehat{R}) = \emptyset$.) Then since $D \subseteq C_0 \cup \widehat{C} \subseteq Z$, D is Z-respecting. By definition, every vertex reachable from X in $G - R_0$ is also reachable from X in $G - D$.

We claim that D is budget-respecting and, furthermore, dominates C_0. Fix a color $i \in [\ell]$; our goal is to prove that $|D \cap E_i| \leq |C_0 \cap E_i|$. To this end, we charge every edge of color i in $D \setminus C_0$ to a distinct edge of color i in $C_0 \setminus D$. Since $D \subseteq C_0 \cup \widehat{C}$, we have that $D \setminus C_0 \subseteq \widehat{C}$, that is, an edge of $D \setminus C_0$ of color i is an edge e_j for some $j \in [\lambda]$ with $e_j \in E_i$ and $e_j \in D \setminus C_0$.

Recall that we are working in the case $A_i = \{j | E_i \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset\}$. Since $e_j \in \widehat{C} \subseteq \widehat{Z}$, we have that $j \in A_i$. Hence, there exists $e'_j \in E_i \cap E(P_j) \cap \widehat{B} \cap C_0$. By the definition of $\widehat{C}$, e_j is the last (closest to Y) edge of $\widehat{B}$ on P_j. Since $e_j \notin C_0$, $e'_j \neq e_j$ and e'_j lies on the subpath of P_j between X and the tail of e_j. This entire subpath is contained in $\widehat{R}$ and, hence, $e'_j \notin D$.

We charge e_j to e'_j. Since $e'_j \in E(P_j) \cap E_i \cap \widehat{B} \cap (C_0 \setminus D)$, for distinct j, the edges e'_j are distinct as the paths P_j do not share an edge belonging to Z and $\widehat{B} \subseteq \widehat{Z} \subseteq Z$. Consequently, $|D \cap E_i| \leq |C_0 \cap E_i|$. This finishes the proof that D dominates C_0.

Since C_0 is important, we have $D = C_0$. In particular, $\widehat{R} \subseteq R_0$. On the other hand, for every $j \in [\lambda]$ we have that $e_j \in \widehat{C} \subseteq \widehat{Z} \subseteq Z \subseteq \bigcup_{i=1}^{\ell} E_i$. In particular, there exists $i \in [\ell]$ such that $e_j \in E_i$ and $j \in A_i$. Hence, we also have $E_i \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset$. But the entire subpath of P_j from X to the tail of e_j lies in $\widehat{R} \subseteq R_0$, while e_j is the last edge of $\widehat{B}$ on P_j. Hence, $e_j \in C_0$. Since the choice of j is arbitrary, $\widehat{C} \subseteq C_0$. Since $\widehat{C}$ is an XY-cut and C_0 is minimal, $\widehat{C} = C_0$ as claimed.

This finishes the proof of Theorem 9. ◀

5 Bound on the number of solutions closest to t

In this section we sketch the proofs of Theorems 3 and 4. The central definition of this section is the following (see also Figure 3).

▶ **Definition 10.** Let G be a directed graph with distinguished vertices s and t and let k be an integer. An *a-bowtie* is a sequence $C_1, C_2, \ldots, C_a$ of pairwise disjoint minimal st-cuts of size k each such that each cut C_i can be partitioned $C_i = A_i \uplus B_i$ such that for every $1 \leq i < j \leq a$, the set A_i is exactly the set of edges of C_i reachable from s in $G - C_j$ and B_j is exactly the set of edges of C_j reachable from s in $G - C_i$.

Our main graph-theoretic result is the following. The proof is deferred to the full version of the paper.

▶ **Theorem 11.** *For every integers $a, k \geq 1$ there exists an integer g such that for every directed graph G with distinguished $s, t \in V(G)$, and a family $\mathcal{U}$ of pairwise disjoint minimal st-cuts of size k each, if $|\mathcal{U}| \geq g$, then $\mathcal{U}$ contains an a-bowtie.*

The next two lemmata are key observations to prove Theorems 3 and 4, respectively, with the help of Theorem 11.

▶ **Lemma 12.** *Let k, g, G, s, t, $\mathcal{F}$, and $\mathcal{G}$ be as in the statement of Theorem 3. Then $\mathcal{G}$ does not contain an a-bowtie for $a > \binom{k+2}{2}$.*

Proof. Assume the contrary, let $(C_i, A_i, B_i)_{i=1}^{a}$ be such a bowtie. Since $a > \binom{k+2}{2}$, there exists $i < j$ with $|A_i| = |A_j|$ and $|B_i| = |B_j|$ (there are $\binom{k+2}{2}$ choices for $(|A_i|, |B_i|)$). However, then $A_i \cup B_j$ and $A_j \cup B_i$ have also cardinality k, are st-cuts, and have together twice the minimum weight. Furthermore, the set of vertices reachable from s in $G - (A_j \cup B_i)$ is a strict superset of the set of vertices reachable from s in $G - C_i$ and $G - C_j$. This contradicts the fact that $C_i, C_j \in \mathcal{G}$. ◀

▶ **Lemma 13.** *Let k, ℓ, $I = (G, s, t, (P_i)_{i=1}^{m}, k)$, $\mathcal{F}$, and $\mathcal{G}$ be as in the statement of Theorem 4. Then $\mathcal{G}$ does not contain a 4-bowtie $(C_i, A_i, B_i)_{i=1}^{4}$ in which the edge set of every path P_j intersects at most one cut C_i.*

Proof. Assume the contrary Let $(C_i, A_i, B_i)_{i=1}^{4}$ be such a 4-bowtie. Consider $i \in \{2, 3\}$ and two edges $e \in A_i$ and $f \in B_i$. In $G - C_4$, the edge e is reachable from s while f is not; consequently, e and f cannot appear on the same input path with e being earlier (by assumption, C_4 is disjoint from the input path in question). A similar reasoning for $G - C_1$ shows that e and f cannot appear on the same input path with f being earlier than e.

Hence, e and f cannot appear together on a single path P_j. For a set of edges D, by the *cost* of D we denote $|\{j | D \cap P_j \neq \emptyset\}|$. Since the choice of e and f was arbitrary, we infer that the sum of costs of $A_2 \cup B_3$ and of $A_3 \cup B_2$ equals the sum of costs of C_2 and of C_3. Hence, both these st-cuts have minimum cost. However, $A_2 \cup B_3$ is closer to t than C_2, a contradiction. ◀

Proof of Theorem 3. Assume $|\mathcal{G}| > g$ for some sufficiently large g to be fixed later. For $i \in [k]$, let $\mathcal{G}^i$ be the set of $u \in \mathcal{G}$ of cardinality i. We apply the Sunflower Lemma to the largest set $\mathcal{G}^i$: If $g > k \cdot k! g_1^k$ for some integer g_1 to be chosen later, there exists $\mathcal{G}_1 \subseteq \mathcal{G}$ with $|\mathcal{G}_1| > g_1$, every element of $\mathcal{G}_1$ being of the same size k', and a set c such that $u \cap v = c$ for every distinct $u, v \in \mathcal{G}_1$.

Let $\widehat{k} = k' - |c|$, $\widehat{u} = u \setminus c$ for every $u \in \mathcal{G}_1$, $\widehat{\mathcal{G}}_1 = \{\widehat{u} \mid u \in \mathcal{G}_1\}$ and $\widehat{G} = G - c$. Since every $u \in \mathcal{U}$ is a minimal st-cut of size k' in G, every $\widehat{u} \in \widehat{\mathcal{G}}_1$ is a minimal st-cut of size $\widehat{k}$ in $\widehat{G}$. Furthermore, every $\widehat{u} \in \widehat{\mathcal{G}}_1$ is a minimal st-cut of size at most $\widehat{k}$ in $\widehat{G}$ of minimum possible weight: if there existed an st-cut $\widehat{x}$ of smaller weight and cardinality at most $\widehat{k}$, then $x = \widehat{x} \cup c$ would be an st-cut in G of cardinality at most k and weight smaller than every element of $\mathcal{G}_1$. Similarly, if there were a minimal st-cut $\widehat{x}$ in $\widehat{G}$ of minimum weight and cardinality at most $\widehat{k}$ that is closer to t than $\widehat{u}$ for some $\widehat{u} \in \widehat{\mathcal{G}}_1$, then $\widehat{x} \cup c$ would be an st-cut in G of cardinality at most k and minimum weight that is closer to t than u, a contradiction. By construction, the elements of $\widehat{\mathcal{G}}_1$ are pairwise disjoint.

Lemma 12 bounds the maximum possible size of a bowtie in $\widehat{\mathcal{G}}_1$. Hence, Theorem 11 asserts that $\widehat{\mathcal{G}}_1$ has size bounded by a function of k. This finishes the proof of the theorem. ◀

Proof of Theorem 4. We proceed similarly as in the proof of Theorem 3, but we need to be a bit more careful with the paths P_j. Assume $|\mathcal{G}| > g$ for some sufficiently large integer g.

As before, we partition $\mathcal{G}$ according to the sizes of elements: for every $i \in [k\ell]$, let $\mathcal{G}^i = \{u \in \mathcal{G} \mid |u| = i\}$. Let $i \in [k\ell]$ be such that $|\mathcal{G}^i| > g/(k\ell)$. For $u \in \mathcal{G}^i$, let $J(u) = \{j \mid u \cap P_j \neq \emptyset\}$. By the assumptions of the theorem, every set $J(u)$ is of cardinality exactly k. We apply the Sunflower Lemma to $\{J(u) \mid u \in \mathcal{G}^i\}$: If $g > (k\ell) \cdot k! \cdot g_1^k$ for some integer g_1 to be fixed later, then there exists $\mathcal{G}_1 \subseteq \mathcal{G}^i$ of size larger than g_1 and a set $I \subseteq [m]$ such that for every distinct $u, v \in \mathcal{G}_1$ we have $J(u) \cap J(v) = I$. For every $u \in \mathcal{G}_1$, let $u_I = u \cap \bigcup_{j \in I} P_j$. Since $|I| \leq k$, there are at most $2^{k\ell}$ choices for u_I among elements $u \in \mathcal{G}_1$. Consequently, there exists $\mathcal{G}_2 \subseteq \mathcal{G}_1$ of cardinality larger than $g_2 := g_1/2^{k\ell}$ such that $u_I = v_I$ for every $u, v \in \mathcal{G}_2$. Denote $c = u_I$ for any $u \in \mathcal{G}_2$.

Let $\widehat{u} := u - c$ for every $u \in \mathcal{G}_2$. Let $\widehat{\mathcal{G}}_2 = \{\widehat{u} \mid u \in \mathcal{G}_2\}$.

Define now $\widehat{G} = G - c$ and define a partition $\widehat{\mathcal{P}}$ of $E(\widehat{G})$ into paths of length at most ℓ as follows: we take all paths P_i for $i \notin I$ and, for every $i \in I$, each edge of $P_i \setminus c$ as a length-1 path. Furthermore, denote $\widehat{k} = k - |I|$. Note that $(\widehat{G}, s, t, \widehat{\mathcal{P}}, \widehat{k})$ is a Chain ℓ-SAT instance for which every $\widehat{u} \in \widehat{\mathcal{G}}_2$ is a solution. Furthermore, $(\widehat{G}, s, t, \widehat{\mathcal{P}}, \widehat{k} - 1)$ is a no-instance, as if $\widehat{x}$ were its solution, then $\widehat{x} \cup c$ would be a solution to $(G, s, t, (P_i)_{i=1}^m, k-1)$, a contradiction. Similarly, if there were a solution $\widehat{x}$ to $(\widehat{G}, s, t, \widehat{\mathcal{P}}, \widehat{k})$ that is closer to t than $\widehat{u}$ for some $\widehat{u} \in \widehat{\mathcal{G}}_2$, then $\widehat{x} \cup c$ would be a solution to $(G, s, t, (P_i)_{i=1}^m, k)$ that is closer to t than u, a contradiction. Furthermore, by construction, the elements of $\widehat{\mathcal{G}}_2$ are pairwise disjoint and no path of $\widehat{\mathcal{P}}$ intersects more than one element of $\widehat{\mathcal{G}}_2$.

Lemma 13 bounds the maximum possible size of a bowtie in $\widehat{\mathcal{G}}_2$. Hence, Theorem 11 asserts that $\widehat{\mathcal{G}}_2$ has size bounded by a function of k and ℓ. This finishes the proof of the theorem. ◀

6 Conclusion

We would like to conclude with a discussion on future research directions. First, our upper bound of $2^{\mathcal{O}(k^2 \log k)}$ on the number of multi-budgeted important separators (Theorem 9) is far from the 4^k bound for the classic important separators. As pointed out by an anonymous reviewer at IPEC 2018, there is an easy lower bound of $k!$ for the number of multi-budgeted important separators: Let $\ell = k$, $k_i = 1$ for every $i \in [\ell]$, and let G consist of k paths from s to t, each path consisting of ℓ edges of different colors. Then there are exactly $k!$ distinct multi-budgeted important separators, as we can freely choose a different color $i \in [\ell]$ to cut on each path. We are not aware of any better lower bound, leaving a significant gap between the lower and upper bounds.

Second, our existential statement of Theorems 3 and 4 can be treated as a weak support of tractability of Chain ℓ-SAT and Weighted st-cut. Are they really FPT when parameterized by the cardinality of the cut?

References

1. Amit Agarwal, Noga Alon, and Moses Charikar. Improved approximation for directed cut problems. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 671–680. ACM, 2007. doi:10.1145/1250790.1250888.
2. Chandra Chekuri, Sudipto Guha, and Joseph Naor. The Steiner k-Cut Problem. *SIAM J. Discrete Math.*, 20(1):261–271, 2006. doi:10.1137/S0895480104445095.
3. Jianer Chen and Iyad A. Kanj. Constrained minimum vertex cover in bipartite graphs: complexity and parameterized algorithms. *J. Comput. Syst. Sci.*, 67(4):833–847, 2003. doi:10.1016/j.jcss.2003.09.003.

4 Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5):21:1–21:19, 2008. doi:10.1145/1411509.1411511.

5 Rajesh Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Marcin Pilipczuk, and Michal Pilipczuk. Designing FPT Algorithms for Cut Problems Using Randomized Contractions. *SIAM J. Comput.*, 45(4):1171–1229, 2016. doi:10.1137/15M1032077.

6 Rajesh Chitnis, László Egri, and Dániel Marx. List H-Coloring a Graph by Removing Few Vertices. *Algorithmica*, 78(1):110–146, 2017. doi:10.1007/s00453-016-0139-6.

7 Rajesh Hemant Chitnis, Marek Cygan, Mohammad Taghi Hajiaghayi, and Dániel Marx. Directed Subset Feedback Vertex Set Is Fixed-Parameter Tractable. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 230–241. Springer, 2012. doi:10.1007/978-3-642-31594-7_20.

8 Rajesh Hemant Chitnis, MohammadTaghi Hajiaghayi, and Dániel Marx. Fixed-Parameter Tractability of Directed Multiway Cut Parameterized by the Size of the Cutset. *SIAM J. Comput.*, 42(4):1674–1696, 2013. doi:10.1137/12086217X.

9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

10 Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3, 2013. doi:10.1145/2462896.2462899.

11 Guy Even, Joseph Naor, Baruch Schieber, and Madhu Sudan. Approximating Minimum Feedback Sets and Multicuts in Directed Graphs. *Algorithmica*, 20(2):151–174, 1998. doi:10.1007/PL00009191.

12 Piotr Faliszewski, Fedor V. Fomin, Daniel Lokshtanov, Dániel Marx, Shmuel Onn, Marcin Pilipczuk, Michał Pilipczuk, Saket Saurabh, and Meirav Zehavi. What's next? Future directions in Parameterized Complexity. *Recent Advances in Parameterized Complexity, Tel-Aviv, Israel*, 2017. Accessed 18.09.2018. URL: https://rapctelaviv.weebly.com/uploads/1/0/5/3/105379375/future.pdf.

13 Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate Max-Flow Min-(Multi)Cut Theorems and Their Applications. *SIAM J. Comput.*, 25(2):235–251, 1996. doi:10.1137/S0097539793243016.

14 Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004. doi:10.1016/S0196-6774(03)00111-1.

15 Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61–71, 2011. doi:10.1016/j.disopt.2010.05.003.

16 Yoichi Iwata. Linear-Time Kernelization for Feedback Vertex Set. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 68:1–68:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.68.

17 Yoichi Iwata, Magnus Wahlström, and Yuichi Yoshida. Half-integrality, LP-branching, and FPT Algorithms. *SIAM J. Comput.*, 45(4):1377–1411, 2016. doi:10.1137/140962838.

18 David R. Karger, Philip N. Klein, Clifford Stein, Mikkel Thorup, and Neal E. Young. Rounding Algorithms for a Geometric Embedding of Minimum Multiway Cut. *Math. Oper. Res.*, 29(3):436–461, 2004. doi:10.1287/moor.1030.0086.

19 Stefan Kratsch and Magnus Wahlström. Representative Sets and Irrelevant Vertices: New Tools for Kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer*

Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012, pages 450–459. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.46.

20 Stefan Kratsch and Magnus Wahlström. Compression via Matroids: A Randomized Polynomial Kernel for Odd Cycle Transversal. *ACM Transactions on Algorithms*, 10(4):20, 2014. doi:10.1145/2635810.

21 Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Parameterized Complexity and Approximability of Directed Odd Cycle Transversal. *CoRR*, abs/1704.04249, 2017. arXiv:1704.04249.

22 Dániel Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006. doi:10.1016/j.tcs.2005.10.007.

23 Dániel Marx. What's Next? Future Directions in Parameterized Complexity. In Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors, *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 469–496. Springer, 2012. doi:10.1007/978-3-642-30891-8_20.

24 Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Transactions on Algorithms*, 9(4):30, 2013. doi:10.1145/2500119.

25 Dániel Marx and Igor Razgon. Fixed-Parameter Tractability of Multicut Parameterized by the Size of the Cutset. *SIAM J. Comput.*, 43(2):355–388, 2014. doi:10.1137/110855247.

26 Marcin Pilipczuk and Magnus Wahlström. Directed multicut is $W[1]$-hard, even for four terminal pairs. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1167–1178. SIAM, 2016. doi:10.1137/1.9781611974331.ch81.

27 Igor Razgon and Barry O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009. doi:10.1016/j.jcss.2009.04.002.