

A Parameterized Algorithm for MIXED-CUT

Ashutosh Rai*

M. S. Ramanujan[†]

Saket Saurabh*

November 23, 2018

Abstract

The classical Menger's theorem states that in any undirected (or directed) graph G , given a pair of vertices s and t , the maximum number of vertex (edge) disjoint paths is equal to the minimum number of vertices (edges) needed to disconnect from s and t . This min-max result can be turned into a polynomial time algorithm to find the maximum number of vertex (edge) disjoint paths as well as the minimum number of vertices (edges) needed to disconnect s from t . In this paper we study a mixed version of this problem, called MIXED-CUT, where we are given an undirected graph G , vertices s and t , positive integers k and l and the objective is to test whether there exist a k sized vertex set $S \subseteq V(G)$ and an l sized edge set $F \subseteq E(G)$ such that deletion of S and F from G disconnects from s and t . We start with a small observation that this problem is NP-complete and then study this problem, in fact a much stronger generalization of this, in the realm of parameterized complexity. In particular we study the MIXED MULTIWAY CUT-UNCUT problem where along with a set of terminals T , we are also given an equivalence relation \mathcal{R} on T , and the question is whether we can delete at most k vertices and at most l edges such that connectivity of the terminals in the resulting graph respects \mathcal{R} . Our main results is a fixed parameter algorithm for MIXED MULTIWAY CUT-UNCUT using the method of recursive understanding introduced by Chitnis et al. (FOCS 2012).

*Institute of Mathematical Sciences, Chennai, India. Emails: {ashutosh|saket}@imsc.res.in

[†]University of Bergen, Norway. Email: M.Ramanujan@uib.no

1 Introduction

Given a graph, a typical *cut problem* asks for finding a set of vertices or edges such that their removal from the graph makes the graph satisfy some separation property. The most fundamental version of the cut problems is MINIMUM CUT, where given a graph and two vertices, called *terminals*, we are asked to find the minimum sized subset of vertices (or edges) of the graph such that deleting them separates the terminals. Minimum cut is can be applied to several problems, which at first do not look like cut problems. Examples include well studied problems such as FEEDBACK VERTEX SET [5] and ODD CYCLE TRANSVERSAL [22]. Even though MINIMUM CUT is known to be in polynomial time for both edge and vertex versions, it becomes NP-hard for many of its generalizations.

Two of the most studied generalizations of MINIMUM CUT problem which are NP-hard are MULTIWAY CUT and MULTICUT. In the MULTIWAY CUT problem, we are given a set of terminals, and we are asked to delete minimum number of vertices (or edges) to separate the terminals from each other. This problem is known to be NP-hard when the number of terminals is at least three. In the MULTICUT problem, given pairs of terminals, we are asked to delete minimum number of vertices (or edges) so that it separates all the given terminal pairs. The MULTICUT problem is known to be NP-hard when the number of pairs of terminals is at least three.

These problems are one of the most well studied problems in all those algorithmic paradigms that are meant for coping with NP-hardness, such as approximation algorithms [1, 9, 11, 12, 14, 21]. The aim of this paper is to look at a generalization of MINIMUM CUT problem in the realm of parameterized complexity. The field of parameterized complexity tries to provide efficient algorithms for NP-complete problems by going from the classical view of single-variate measure of the running time to a multi-variate one. It aims at getting algorithms of running time $f(k)n^{\mathcal{O}(1)}$, where k is an integer measuring some aspect of the problem. These algorithms are called fixed parameter tractable (FPT) algorithms and the integer k is called the *parameter*. In most of the cases, the solution size is taken to be the parameter, which means that this approach gives faster algorithms when the solution is of small size. For more background on parameterized complexity, the reader is referred to the monographs [8, 10, 20].

Cut problems were looked at under the realm of Parameterized Complexity by Marx [17] for the first time, who showed that MULTIWAY CUT is FPT when parameterized by the solution size and MULTICUT is FPT when parameterized by the solution size plus the number of terminals. Subsequently, a lot of work has been done on cut problems in the field of parameterized complexity [3, 4, 7, 15, 16, 18, 19]. Recently, Chitnis et al. [6] introduced the technique of *randomized contractions* and used that to solve the UNIQUE LABEL COVER problem. They also show that the same techniques can be applied to solve a generalization of MULTIWAY CUT problem, namely MULTIWAY CUT-UNCUT, where an equivalence relation \mathcal{R} is also supplied along with the set of terminals and we are to delete minimum number of vertices (or edges) such that the terminals lie in the same connected of the resulting graph if and only if they lie in the same equivalence class of \mathcal{R} .

Another interesting generalization of the MINIMUM CUT problem is the one where we allow both vertices and edges to be deleted to achieve the desired separation properties. In an instance of MIXED CUT problem, we are asked whether we can separate the two given terminals by deleting at most k vertices and at most l edges. The problem can also be looked at as asking to delete k vertices so that there are at most l edge disjoint paths between the two terminals.

Even though the vertex and edge versions of MINIMUM CUT problem are polynomial time solvable, we show that allowing deletion of both, the vertices and the edges, makes the MIXED CUT problem NP-hard. To show that, we use a simple reduction from the BIPARTITE PARTIAL VERTEX COVER problem which was recently shown to be NP-hard [2, 13]. Hence, MIXED CUT

becomes interesting from the parameterized complexity point of view.

The MIXED CUT problem can also be generalized in the same way as MINIMUM CUT in some cases. In the MIXED MULTIWAY CUT problem, which is an analogue of MULTIWAY CUT, given a set of terminals, we are asked to delete at most k vertices and l edges to separate the terminals from each other. Similarly, we can define MIXED MULTIWAY CUT-UNCUT (MMCU) as an analogue of MULTIWAY CUT-UNCUT problem where we have to preserve the connectivity between the terminals which belong to the same equivalence class, while separating the terminals which belong to different equivalence classes, by deleting at most k vertices and at most l edges. We defer the formal definition of the problem to the next section. It is easy to see that MMCU not only generalizes MIXED CUT and MIXED MULTIWAY CUT, but also both edge and vertex versions of MULTIWAY CUT and MULTIWAY CUT-UNCUT problems.

Studying the parameterized complexity of MMCU is the main focus of this paper. We show that MMCU is FPT when parameterized by the solution size, i.e. $k + l$. By doing so, we show that MIXED CUT is also FPT when parameterized by the solution size.

We use the irrelevant vertex technique introduced by Chitnis et al. [6] to solve the problem. The main observation is that if there is a small vertex separation which divides the graph into big parts, then we can recursively reduce the size of one of the big parts. Otherwise, the graph is highly connected, and the structure of the graph can be exploited to obtain a solution. The techniques and the presentation of the paper are heavily borrowed from that of [6], but we do need many technical modifications to make the algorithm work for MMCU. Allowing edge deletions makes us modify many of the initial operations on the graph. The algorithm in [6] returns the minimum solutions in the recursive steps. Since we allow both edge and vertex deletion, there is no clear ordering on the solutions, and hence we need to look for solutions of all possible sizes while making the recursive call.

2 Preliminaries

In this section, we first give the notations and definitions which are used in the paper. Then we state some basic properties of mixed-cuts and some known results which will be used later in the paper.

Notations and Definitions: For a graph G , we denote the set of vertices of the graph by $V(G)$ and the set of edges of the graph by $E(G)$. We denote $|V(G)|$ and $|E(G)|$ by n and m respectively, where the graph is clear from context. For a set $S \subseteq V(G)$, the *subgraph of G induced by S* is denoted by $G[S]$ and it is defined as the subgraph of G with vertex set S and edge set $\{(u, v) \in E(G) : u, v \in S\}$ and the subgraph obtained after deleting S is denoted as $G - S$. For $F \subseteq E(G)$, by $V(F)$ we denote the set $\{v \mid \exists u \text{ such that } uv \in F\}$. For a set $Z = V' \cup E'$ where $V' \subseteq V(G)$ and $E' \subseteq E(G)$, by $G(Z)$ we denote the subgraph $G' = (V' \cup V(E'), E')$. For a tuple $\mathcal{X} = (X, F)$ such that $X \subseteq V(G)$ and $F \subseteq E(G)$, by $G - \mathcal{X}$ we denote the graph $G' = (V(G) \setminus X, E(G) \setminus F)$ and by $V(\mathcal{X})$ we denote the vertex set $X \cup V(E)$. All vertices adjacent to a vertex v are called neighbours of v and the set of all such vertices is called *open neighbourhood* of v , denoted by $N_G(v)$. For a set of vertices $S \subseteq V(G)$, we define $N_G(S) = (\cup_{v \in S} N(v)) \setminus S$. We drop the subscript G when the graph is clear from the context.

We define the MIXED CUT and MIXED MULTIWAY CUT-UNCUT problems as follows.

MIXED CUT

Input: A multigraph G , vertices $s, t \in V(G)$, integers k and l .

Parameters: k, l

Question: Does there exist $X \subseteq V(G)$ and $F \subseteq E(G)$ such that $|X| \leq k$, $|F| \leq l$ and s and t are in different connected components of $G - (X, F)$?

MIXED MULTIWAY CUT-UNCUT (MMCUC)

Input: A multigraph G , a set of terminals $T \subseteq V(G)$, and equivalence relation \mathcal{R} on the set T and integers k and l .

Parameters: k, l

Question: Does there exist $X \subseteq (V(G) \setminus T)$ and $F \subseteq E(G)$ such that $|X| \leq k$, $|F| \leq l$ and for all $u, v \in T$, u and v belong to the same connected component of $G - (X, F)$ if and only if $(u, v) \in \mathcal{R}$?

We say that a tuple $\mathcal{X} = (X, F)$, where $X \subseteq V(G) \setminus T$ and $F \subseteq E(G)$, is a solution to a MMCUC instance $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ if $|X| \leq k$, $|F| \leq l$ and for all $u, v \in T$, u and v belong to the same connected component of $G - (X, F)$ if and only if $(u, v) \in \mathcal{R}$. We define a partial order on the solutions of the instance \mathcal{I} . For two solutions $\mathcal{X} = (X, F)$ and $\mathcal{X}' = (X', F')$ of a MMCUC instance \mathcal{I} , we say that $\mathcal{X}' \leq \mathcal{X}$ if $X' \subseteq X$ and $F' \subseteq F$. We say that a solution \mathcal{X} to an MMCUC instance \mathcal{I} is *minimal* if there does not exist another solution \mathcal{X}' to \mathcal{I} such that $\mathcal{X}' \neq \mathcal{X}$ and $\mathcal{X}' \leq \mathcal{X}$. For a solution $\mathcal{X} = (X, F)$ of an MMCUC instance $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ and $v \subseteq V(G)$, we say that \mathcal{X} *affects* v if either $v \in X$ or there exists $u \in V(G)$ such that $uv \in F$.

Observation 1. *If $\mathcal{X} = (X, F)$ is a minimal solution to a MMCUC instance $\mathcal{I} = (G, T, \mathcal{R}, k, l)$, then none of the edges in F are incident to X .*

Lemma 1 ([6]). *Given a set U of size n together with integers $0 \leq a, b \leq n$, one can in $\mathcal{O}(2^{\mathcal{O}(\min(a,b) \log(a+b))} n \log n)$ time construct a family \mathcal{F} of at most $\mathcal{O}(2^{\mathcal{O}(\min(a,b) \log(a+b))} \log n)$ subsets of U , such that the following holds: for any sets $A, B \subseteq U$, $A \cap B = \emptyset$, $|A| \leq a$, $|B| \leq b$, there exists a set $S \in \mathcal{F}$ with $A \subseteq S$ and $B \cap S = \emptyset$.*

Definition 1 ([6]). *Let G be a connected graph and $V^\infty \subseteq V(G)$ a set of undeletable vertices. A triple (Z, V_1, V_2) of subsets of $V(G)$ is called a (q, k) -good node separation, if $|Z| \leq k$, $Z \cap V^\infty = \emptyset$, V_1 and V_2 are vertex sets of two different connected components of $G - Z$ and $|V_1 \setminus V^\infty|, |V_2 \setminus V^\infty| > q$.*

Definition 2 ([6]). *Let G be a connected graph, $V^\infty \subseteq V(G)$ a set of undeletable vertices, and $T_b \subseteq V(G)$ a set of border terminals in G . A pair $(Z, (V_i)_{i=1}^l)$ is called a (q, k) -flower separation in G (with regard to border terminals T_b), if the following holds:*

- $1 \leq |Z| \leq k$ and $Z \cap V^\infty = \emptyset$; the set Z is the core of the flower separation $(Z, (V_i)_{i=1}^l)$;
- V_i are vertex sets of pairwise different connected components of $G - Z$, each set V_i is a petal of the flower separation $(Z, (V_i)_{i=1}^l)$;
- $V(G) \setminus (Z \cup \bigcup_{i=1}^l V_i)$, called a stalk, contains more than q vertices of $V \setminus V^\infty$;
- for each petal V_i we have $V_i \cap T_b = \emptyset$, $|V_i \setminus V^\infty| \leq q$ and $N_G(V_i) = Z$;
- $|\bigcup_{i=1}^l V_i \setminus V^\infty| > q$.

Lemma 2 ([6]). *Given a connected graph G with undeletable vertices $V^\infty \subseteq V(G)$ and integers q and k , one may find in $\mathcal{O}(2^{\mathcal{O}(\min(q,k) \log(q+k))} n^3 \log n)$ time a (q, k) -good node separation of G , or correctly conclude that no such separation exists.*

Lemma 3 ([6]). *Given a connected graph G with undeletable vertices $V^\infty \subseteq V(G)$ and border terminals $T_b \subseteq V(G)$ and integers q and k , one may find in $\mathcal{O}(2^{\mathcal{O}(\min(q,k) \log(q+k))} n^3 \log n)$ time a (q, k) -flower separation in G w.r.t. T_b , or correctly conclude that no such flower separation exists.*

Lemma 4 ([6]). *If a connected graph G with undeletable vertices $V^\infty \subseteq V(G)$ and border terminals $T_b \subseteq V(G)$ does not contain a (q, k) -good node separation or a (q, k) -flower separation w.r.t. T_b then, for any $Z \subseteq V(G) \setminus V^\infty$ of size at most k , the graph $G - Z$ contains at most $(2q + 2)(2^k - 1) + |T_b| + 1$ connected components containing a vertex of $V(G) \setminus V^\infty$, out of which at most one has more than q vertices not in V^∞ .*

3 NP-completeness of Mixed Cut

We prove that MIXED CUT is NP-complete by giving a reduction from the BIPARTITE PARTIAL VERTEX COVER problem which is defined as follows.

BIPARTITE PARTIAL VERTEX COVER (BPVC)

Input: A bipartite graph $G = (X \uplus Y, E)$, integers p and q

Output: Does there exist $S \subseteq V(G)$ such that $|S| \leq p$ and at least q edges in E are incident on X ?

Theorem 1 ([2, 13]). *BPVC is NP-complete.*

For an instance of BPVC, we assume that the given bipartite graph does not have any isolated vertices, as a reduction rule can be applied in polynomial time which takes care of isolated vertices and produces an equivalent instance. Given an instance (G, p, q) of BPVC where $G = (X \uplus Y, E)$ is a bipartite graph, we get an instance (G', s, t, k, l) of MIXED CUT as follows. To get the graph G' , we introduce two new vertices s and t and add all edges from s to X and t to Y . More formally, $G' = (V', E')$ where $V' = V(G) \cup \{s, t\}$ and $E' = E \cup \{sx \mid x \in X\} \cup \{ty \mid y \in Y\}$. Then we put $k = p$ and $l = m - q$, where $m = |E|$. It is easy to see that (G, p, q) is a YES instance of BPVC if and only if (G', s, t, k, l) is a YES instance of MIXED CUT, and hence we get the following theorem.

Theorem 2. *MIXED CUT is NP-complete even on bipartite graphs.*

4 An algorithm for MMCU

In this section, we describe the FPT algorithm for MMCU. To that end, we first describe how we can bound the number of equivalence classes in a connected component of the given graph.

4.1 Reduction of Equivalence Classes

We first show that if there exists a vertex which has a large number of vertex disjoint paths to terminals which are in different equivalence classes, then that vertex has to be part of the solution.

Lemma 5. *Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be a MMCU instance and let $v \in V(G) \setminus T$. Assume that there exist $k + l + 2$ paths $P_1, P_2, \dots, P_{k+l+2}$ in G , such that:*

- *for each $1 \leq i \leq k + l + 2$, the path P_i is a simple path that starts at v and ends at $v_i \in T$;*
- *any two paths P_i and P_j , $i \neq j$, are vertex disjoint except for the vertex v .*
- *for any $i \neq j$, $(v_i, v_j) \notin \mathcal{R}$.*

Then for any solution (X, F) of \mathcal{I} we have $v \in X$.

Proof. For a contradiction assume that $v \notin X$. Since for $i \neq j$, any two paths P_i and P_j are vertex disjoint except for the vertex v , we have that after we delete vertices of X and edges of F there are still two paths remaining among the given collection, say P_a and P_b . But then we get a path from some $v_a \in T$ to some vertex $v_b \in T$ in $G - (X, F)$ by concatenating P_a and P_b such that $(v_a, v_b) \notin \mathcal{R}$. This contradicts the fact that (X, F) is a solution to \mathcal{I} . \square

Lemma 6. *Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be a MMCU instance. For any $v \in V(G)$, we can verify if v satisfies conditions of Lemma 5 in $\mathcal{O}((k+l)n^2)$ time.*

Proof. To verify whether a vertex $v \in V(G)$ satisfies the conditions of Lemma 5 we do as follows. Given a graph G we transform it into a directed network D as follows. For every edge $uv \in E(G)$ we have two directed arcs (u, v) and (v, u) in $E(D)$ and for every equivalence class $Z \in \mathcal{R}$ we make a new vertex t_z and give a directed arc from $z \in Z$ to t_z (that is we add (z, t_z)). Finally, we add a super-sink t and give directed arc (t_z, t) for every $Z \in \mathcal{R}$. Furthermore, give every arc unit capacity. Now to check whether v satisfies the conditions of Lemma 5 all we need to check whether there is a flow of size $k + l + 2$ in D from v to t . This can be done by running $k + l + 2$ rounds of augmenting path and thus the running time is upper bounded by $\mathcal{O}((k + l + 2) \cdot (|E(D)| + |V(D)|))$. Thus the claimed running time follows. \square

Step 1. *For each $v \in V(G)$, if v satisfies the conditions of Lemma 5, delete v from the graph and decrease k by one; if k becomes negative by this operation, return NO. Then restart the algorithm.*

Using the algorithm described in Lemma 6, each application of Step 1 takes $\mathcal{O}((k+l)n^3)$ time. As Step 1 can not be applied more than k times, all applications of the step take $\mathcal{O}(k(k+l)n^3)$ time. Now we show how application of Step 1 gives a bound on number of equivalence classes.

Lemma 7. *Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be the instance of MMCU which we get after the exhaustive application of Step 1. If there exists a connected component in G containing terminals from more than $(k+l)(k+l+1)$ equivalence classes of \mathcal{R} , then \mathcal{I} is a NO instance of MMCU.*

Proof. For a contradiction assume that \mathcal{I} is a YES instance of MMCU. We will show that if this is the case then there exists an opportunity to apply Step 1. Let (X, F) be a solution to \mathcal{I} and X' be a set of vertices containing all of X and exactly one vertex from each of the edges in F . Clearly, the size of X' is at most $k + l$. Let C_1, \dots, C_s be the connected components of $G - X'$ that contains at least one terminal vertices. Since G contains terminals from more than $(k+l)(k+l+1)$ equivalence classes of \mathcal{R} and each connected component of $G - X'$ contains vertices from at most one equivalence class we have that $s > (k+l)(k+l+1)$. Thus by Pigeon Hole Principle there exists a vertex $v \in X'$ that has neighbors in at least $k + l + 2$ connected components. These connected components together with v imply that we could apply Step 1. This contradicts our assumption that on \mathcal{I} Step 1 is no longer applicable. Thus indeed \mathcal{I} is a NO instance of MMCU. \square

Step 2. *If there exist $u, v \in T$ such that u and v lie in different connected components of G or there exists a connected component of G with terminals of more than $(k+l)(k+l+1)$ equivalence classes of \mathcal{R} , return NO.*

Correctness of Step 2 follows from Lemma 7 and it can be applied in $\mathcal{O}(n^2)$ time. In the next step, take care of disconnected graphs which helps us assume that the input graph is connected.

Step 3. *Let C_1, C_2, \dots, C_t be the vertices corresponding to the connected components of the graph G in the input instance $\mathcal{I} = (G, T, \mathcal{R}, k, l)$. For each C_i and for every $k' \in \{0, 1, \dots, k\}$ and $l' \in \{0, 1, \dots, l\}$ we pass the instance $\mathcal{I}_{k', l'}^i = (G, T \cap C_i, \mathcal{R}|_{T \cap C_i}, k', l')$ to the next step. If for every $i \in \{1, \dots, t\}$ there exists $\mathcal{I}_{k'_i, l'_i}^i$ such that the subroutine returns YES for $\mathcal{I}_{k'_i, l'_i}^i$ and $\sum_{i \in \{1, \dots, t\}} k'_i \leq k$, $\sum_{i \in \{1, \dots, t\}} l'_i \leq l$, then return YES, otherwise return NO.*

The correctness of this step is easy to see, as after the application of Step 2, each equivalence class is confined to at most one connected component of the graph. The step can be applied in $\mathcal{O}(n^2)$ time and gives rise to $\mathcal{O}(kln)$ subproblems. After application of Step 3, we can assume that G is connected and that the number of equivalence classes in G are bounded by $(k+l)(k+l+1)$.

4.2 Operations on the Graph

Definition 3. Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be an MMCU instance and let $v \in V(G) \setminus T$. By bypassing a vertex v we mean the following operation: we delete the vertex v from the graph and, for any $u_1, u_2 \in N_G(v)$, we add an edge (u_1, u_2) if it is not already present in G .

Definition 4. Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be an MMCU instance and let $u, v \in T$. By identifying vertices u and v , we mean the following operation: we make a new set $T' = (T \setminus \{u, v\}) \cup \{x_{uv}\}$ and for each edge of the form $uw \in E(G)$ or $vw \in E(G)$, we add an edge $x_{uv}w$ to $E(G)$. Observe that the operation might add parallel edges.

Lemma 8. Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be a MMCU instance, let $v \in V(G) \setminus T$ and let $\mathcal{I}' = (G', T, \mathcal{R}, k, l)$ be the instance \mathcal{I} with v bypassed. Then:

- if $\mathcal{X} = (X, F)$ is a solution to \mathcal{I}' , then \mathcal{X} is a solution to \mathcal{I} as well;
- if $\mathcal{X} = (X, F)$ is a solution to \mathcal{I} and $v \notin X$ and for all $u \in N(v)$ $vu \notin F$ then \mathcal{X} is a solution to \mathcal{I}' as well.

Proof. We first prove the first item in the lemma. Suppose $\mathcal{X} = (X, F)$ is a solution to \mathcal{I}' but it is not a solution to \mathcal{I} . This implies that either there exists $(x, y) \in \mathcal{R}$ and x and y belong to two different connected components of $G - \mathcal{X}$ or $(x, y) \notin \mathcal{R}$ and they are in the same connected component of $G - \mathcal{X}$. In the first case we know that x and y belong to the same connected component of $G' - \mathcal{X}$ and thus we know that the two connected components of $G - \mathcal{X}$ which contain x and y has neighbors of v . This contradicts that x and y belong to two different connected components of $G - \mathcal{X}$. For the later case if there is a path containing x and y then it must contain v but then in G' we have an edge between the end-points of v on this path, which in turn would imply a path between x and y in $G' - \mathcal{X}$. One can prove the second statement along the similar lines. \square

Lemma 9. Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be a MMCU instance and let $u, v \in T$ be two different terminals with $(u, v) \notin \mathcal{R}$, such that $uv \in E(G)$, then for any solution $\mathcal{X} = (X, F)$ of \mathcal{I} , we have $uv \in F$.

The proof of the Lemma 9 follows from the fact that any solution must delete the edge uv to disconnect u from v . The proof of the next lemma follows by simple observation that u and v have at least $k+l+1$ internally vertex disjoint paths or from the fact that $(u, v) \in \mathcal{R}$ and thus after deleting the solution they must belong to the same connected component and thus every minimal solution does not use the edge $uv \in E(G)$.

Lemma 10. Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be a MMCU instance and let $u, v \in T$ be two different terminals with $(u, v) \in \mathcal{R}$, such that $uv \in E(G)$ or $|N_G(u) \cap N_G(v)| > k+l$. Let \mathcal{I}' be instance \mathcal{I} with terminals u and v identified. Then set of minimal solutions of \mathcal{I} and \mathcal{I}' is the same.

Lemma 11. Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be an MMCU instance and let $V' = \{v_1, v_2, \dots, v_t\} \subseteq T$ be different terminals of the same equivalence class of \mathcal{R} , pairwise nonadjacent and such that $N_G(u_1) = N_G(u_2) = \dots = N_G(u_t) \subseteq V(G) \setminus T$ and $t > l+2$. Let \mathcal{I}' be obtained from \mathcal{I} by deleting all but $l+2$ terminals in U (and all pairs that contain the deleted terminals in \mathcal{R}). Then the set of minimal solutions to \mathcal{I} and \mathcal{I}' are equal.

Proof. Without loss of generality, let the set deleted terminals be $U' := \{u_{l+3}, \dots, u_t\}$ and let $N := N_G(u_1) = N_G(u_2) = \dots = N_G(u_t)$. Let $u, v \in V(G) \setminus U'$. We claim that for any minimal solution $\mathcal{X} = (X, F)$ of \mathcal{I} , u and v are in the same connected component of $G - \mathcal{X}$ if and only if they are in the same connected component of $G' - \mathcal{X}$. The backward direction is trivial. To show the forward direction, we just need to show that F does not contain any edges incident on U , because then if a path visits a vertex in U' , then we can redirect it via one of u_i 's which remain in the graph. For $v \in X \cap N$, by Observation 1, we have that F does not contain any edges incident on v . For $v \in N \setminus X$, there are $u_a, u_b \in U \setminus U'$ which are adjacent to v in $G - \mathcal{X}$. This gives us that v lies in the same connected component of $G - \mathcal{X}$ as U , and hence a minimal solution does not contain any of the edges between x and U . This shows that \mathcal{X} is a minimal solution for \mathcal{I}' as well.

For the converse, let \mathcal{X} be a minimal solution to \mathcal{I}' . We just need to show that if $u, v \in T$ such that $v \in T \setminus U'$ and $u \in U'$, they lie in same connected component of $G - \mathcal{X}$ if and only if $(u, v) \in \mathcal{R}$. Since $l \geq 0$, we have that $|U \setminus U'| \geq 2$. Now it is easy to see that $N \subsetneq X$ because otherwise the vertices of $U \setminus U'$ will not be in the same connected component of $G' - \mathcal{X}$. Hence, there exists $x \in N \setminus X$. Since $|N_{G'}(X) \cap U'| = l + 2$, v is adjacent to at least 2 vertices of U' in $G' - \mathcal{X}$. Without loss of generality, let $u_1, u_2 \in U \setminus U'$ be these two vertices. Now let $v \in T \setminus U'$ and $u \in U'$. If $(v, u_1) \in \mathcal{R}$, then v and x belong to the same connected component of $G' - \mathcal{X}$ and hence v and u belong to same connected component of $G - \mathcal{X}$. Similarly if $(v, u_1) \notin \mathcal{R}$, then v and x belong to the different connected components of $G' - \mathcal{X}$ and hence v and u belong to different connected components of $G - \mathcal{X}$. This concludes the proof of the lemma. \square

Lemma 12. *Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be an MMCU instance and let $uv \in E(G)$ be an edge with multiplicity more than $l + 1$. Then for any minimal solution $\mathcal{X} = (X, F)$ of \mathcal{I} , F does not contain any copies of uv .*

Proof. If $\{u, v\} \cap X \neq \emptyset$, then by Observation 1, we have that none of the copies of uv are in F . Otherwise, F contains at most l copies of edge uv . Let $\mathcal{X}' = (X, F \setminus \{uv\})$. Then we have that for any two $x, y \in V(G)$, x and y are adjacent in $G - \mathcal{X}$ if and only if they are adjacent in $G - \mathcal{X}'$, contradicting the minimality of \mathcal{X} . \square

a minimal solution for \mathcal{I}' as well.

4.3 Borders and Recursive Understanding

In this section, we define the bordered problem and describe the recursive phase of the algorithm. Let $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ be an MMCU instance and let $T_b \subseteq V(G) \setminus T$ be a set of border terminals, where $|T_b| \leq 2(k + l)$. Define $\mathcal{I}_b = (G, T, \mathcal{R}, k, l, T_b)$ to be an instance of the bordered problem. By $\mathbb{P}(\mathcal{I}_b)$ we define the set of all tuples $\mathcal{P} = (X_b, E_b, \mathcal{R}_b, k', l')$, such that $X_b \subseteq T_b$, E_b is an equivalence relation on $T_b \setminus X_b$, \mathcal{R}_b is an equivalence relation on $T \cup (T_b \setminus X_b)$ such that $E_b \subseteq \mathcal{R}_b$ and $\mathcal{R}_b|_T = \mathcal{R}$, $k' \leq k$ and $l' \leq l$. For a tuple $\mathcal{P} = (X_b, E_b, \mathcal{R}_b, k', l')$, by $G_{\mathcal{P}}$ we denote the graph $G \cup E_b$, that is the graph G with additional edges E_b .

We say that a tuple $\mathcal{X} = (X, F)$ is a solution to $(\mathcal{I}_b, \mathcal{P})$ where $\mathcal{P} = (X_b, E_b, \mathcal{R}_b, k', l')$ if $|X| \leq k'$, $|F| \leq l'$ and for all $u, v \in T \cup (T_b \setminus X_b)$, u and v belong to the same connected component of $G_{\mathcal{P}} - (X, F)$ if and only if $(u, v) \in \mathcal{R}_b$. We also say that \mathcal{X} is a solution to $\mathcal{I}_b = (G, T, \mathcal{R}, k, l, T_b)$ whenever \mathcal{X} is a solution to $\mathcal{I} = (G, T, \mathcal{R}, k, l)$. Now we define the bordered problem as follows.

BORDER-MIXED MULTIWAY CUT-UNCUT(B-MMCU)

Input: An MMCU instance $\mathcal{I} = (G, T, \mathcal{R}, k, l)$ with G being connected and a set $T_b \subseteq V(G) \setminus T$ such that $|T_b| \leq 2(k+l)$; denote $\mathcal{I}_b = (G, T, \mathcal{R}, k, l, T_b)$.
Output: For each $\mathcal{P} = (X_b, E_b, \mathcal{R}_b, k', l') \in \mathbb{P}(\mathcal{I}_b)$, output a $\text{sol}_{\mathcal{P}} = \mathcal{X}_{\mathcal{P}}$ being a minimal solution to $(\mathcal{I}_b, \mathcal{P})$, or $\text{sol}_{\mathcal{P}} = \perp$ if no solution exists.

It is easy to see that MMCU reduces to B-MMCU, by putting $T_b = \emptyset$. Also, in this case, any answer to B-MMCU for $\mathcal{P} = (\emptyset, \emptyset, \mathcal{R}, k, l)$ returns a solution for MMCU instance. To bound the size of the solutions returned for an instance of B-MMCU we observe the following.

$$\begin{aligned} |\mathbb{P}(\mathcal{I}_b)| &\leq (k+1)(l+1)(1 + |T_b|(|T_b| + (k+l)(k+l+1)))^{|T_b|} \\ &\leq (k+1)(l+1)(1 + 2(k+l)^2(k+l+3))^{2(k+l)} \\ &= 2^{\mathcal{O}((k+l) \log(k+l))} \end{aligned}$$

This is true because \mathcal{R}_b has at most $(k+l)(k+l+1) + |T_b|$ equivalence classes, E_b has at most T_b equivalence classes, each $v \in T_b$ can either go to X_b or choose an equivalence class in \mathcal{R}_b and E_b , and k' and l' have $k+1$ and $l+1$ possible values respectively. Let $q = (k+2l)(k+1)(l+1)(1+2(k+l)^2(k+l+3))^{2(k+l)} + k+l$, then all output solutions to a B-MMCU instance \mathcal{I}_b affect at most $q - (k+l)$ vertices in total. Now we are ready to prove the lemma which is central for the recursive understanding step.

Lemma 13. *Assume we are given a B-MMCU instance $\mathcal{I}_b = (G, T, \mathcal{R}, k, l, T_b)$ and two disjoint sets of vertices $Z, V^* \subseteq V(G)$, such that $|Z| \leq k+l$, $Z \cap T = \emptyset$, $Z_W := N_G(V^*) \subseteq Z$, $|V^* \cap T_b| \leq k+l$ and the subgraph of G induced by $W := V^* \cup Z_W$ is connected. Denote $G^* = G[W]$, $T_b^* = (T_b \cup Z_W) \cap W$, $T^* = T \cap W$, $\mathcal{R}^* = \mathcal{R}|_{T \cap W}$ and $\mathcal{I}^* = (G^*, T^*, \mathcal{R}^*, k, l, T_b^*)$. Then \mathcal{I}^* is a proper B-MMCU instance. Moreover, if we denote by $(\text{sol}_{\mathcal{P}^*}^*)_{\mathcal{P}^* \in \mathbb{P}(\mathcal{I}_b^*)}$ an arbitrary output to the B-MMCU instance \mathcal{I}_b^* and*

$$U(\mathcal{I}_b^*) = T_b^* \cup \{v \in V(G) \mid \mathcal{P}^* \in \mathbb{P}(\mathcal{I}_b^*), \text{sol}_{\mathcal{P}^*}^* = \mathcal{X}_{\mathcal{P}^*}^* \neq \perp \text{ and } \mathcal{X}_{\mathcal{P}^*}^* \text{ affects } v\},$$

then there exists a correct output $(\text{sol}_{\mathcal{P}})_{\mathcal{P} \in \mathbb{P}(\mathcal{I}_b)}$ to the B-MMCU instance \mathcal{I}_b such that whenever $\text{sol}_{\mathcal{P}} = \mathcal{X}_{\mathcal{P}} \neq \perp$ and $\mathcal{X}_{\mathcal{P}}$ is a minimal solution to $(\mathcal{I}_b, \mathcal{P})$ then $V(\mathcal{X}_{\mathcal{P}}) \cap V^ \subseteq U(\mathcal{I}_b^*)$.*

Proof. To see that \mathcal{I}_b^* is a proper B-MMCU instance, we observe that $G^* = G[W]$ is connected, $|Z_W| \leq |Z| \leq k+l$ and $|V^* \cap T_b| \leq k+l$, and hence $|T_b^*| \leq 2(k+l)$.

To show the second part of the lemma, let us assume $\mathcal{P} = (X_b, E_b, \mathcal{R}_b, k', l') \in \mathbb{P}(\mathcal{I}_b)$. Let $\mathcal{X}_{\mathcal{P}} = (X, F)$ be a solution to $(\mathcal{I}_b, \mathcal{P})$. We show that there exists another solution $\mathcal{X}'_{\mathcal{P}} = (X', F')$ to $(\mathcal{I}_b, \mathcal{P})$ such that $|X'| \leq |X|$, $|F'| \leq |F|$ and $V(\mathcal{X}'_{\mathcal{P}}) \cap V^* \subseteq U(\mathcal{I}_b^*)$. To that end, we define a tuple $\mathcal{P}^* = (X_b^*, E_b^*, \mathcal{R}_b^*, k^*, l^*)$ as follows. Let $F'' := F \setminus E(G[W])$, $X'' := (X \setminus V^*)$ and $\mathcal{X}''_{\mathcal{P}} = (X'', F'')$. Let $G'_{\mathcal{P}}$ be the graph such that $V(G'_{\mathcal{P}}) = V(G) \setminus W$ and $E(G'_{\mathcal{P}}) = E(G_{\mathcal{P}} - V^*) \setminus E(G_{\mathcal{P}}[Z_W])$. Observe that $V(G^*) \cap V(G'_{\mathcal{P}}) = Z_W$ and $E(G^*) \cap E(G'_{\mathcal{P}}) = \emptyset$.

- $X_b^* = X \cap T_b^*$.
- E_b^* is an equivalence relation on $T_b^* \setminus X_b^*$ such that for $(u, v) \in E_b^*$ if and only if u and v are in the same connected component of $G'_{\mathcal{P}} - \mathcal{X}''_{\mathcal{P}}$.
- \mathcal{R}_b^* is an equivalence relation on $T^* \cup (T_b^* \setminus X_b^*)$ such that $(u, v) \in \mathcal{R}_b^*$ if and only if u and v are in the same connected component of $G_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$.
- $k^* := |X \cap W|$ and $l^* := |F \cap E(G[W])|$.

Since $|X| \leq k' \leq k$ and $|F| \leq l' \leq l$, we have that $k^* \leq k$ and $l^* \leq l$. Also, $E_b^* \subseteq \mathcal{R}_b^*$ because they are both corresponding to relation of being in the same connected component, but the graph considered for defining E_b^* is a subgraph of the one considered for defining \mathcal{R}_b^* . Hence we have that $\mathcal{P}^* \in \mathbb{P}(\mathcal{I}_b^*)$. We claim that $(X \cap W, F \cap E[(G[W])])$ is a solution to $(\mathcal{I}_b^*, \mathcal{P}^*)$. By definition, we have that $|X \cap W| \leq k^*$ and $|F \cap E[(G[W])]| \leq l^*$. Also, by definition of X_b^* , we have that $X \cap W \cap T_b^* = X_b^*$. Now look at two vertices $u, v \in T^* \cup (T_b^* \setminus X_b^*)$. By definition, we have that $(u, v) \in \mathcal{R}_b^*$ if and only if there exists a path P connecting u and v in $G_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$. We claim that such a path P exists if and only if there exists a path P^* connecting u and v in $G_{\mathcal{P}^*} - (X \cap W, F \cap E[(G[W])])$. It is indeed so, because subpath of P with internal vertices in $V(G) \setminus W$ corresponds to an edge in E_b^* and vice versa. Thus, u and v are in the same connected component of $G_{\mathcal{P}^*} - (X \cap W, F \cap E[(G[W])])$ if and only if $(u, v) \in \mathcal{R}_b^*$ and hence $(X \cap W, F \cap E[(G[W])])$ is a solution to $(\mathcal{I}_b^*, \mathcal{P}^*)$.

So we get that $\text{sol}_{\mathcal{P}^*}^* = \mathcal{X}_{\mathcal{P}^*}^* =: (X^*, F^*) \neq \perp$ and $|X^*| \leq k^*$ and $|F^*| \leq l^*$. Let $\mathcal{X}'_{\mathcal{P}} = (X', F')$, where $X' = (X \setminus W) \cup X^*$ and $F' = (F \setminus E[(G[W])]) \cup F^*$. We claim that $\mathcal{X}'_{\mathcal{P}}$ is a solution for $(\mathcal{I}_b, \mathcal{P})$. As $|X^*| \leq |X \cap W|$ and $|F^*| \leq |F \cap E[(G[W])]|$, we have that $|X'| \leq |X| \leq k'$ and $|F'| \leq |F| \leq l'$. Also, since $X_b^* = X \cap T_b^*$ by definition of X_b^* and by properties of $\mathcal{X}_{\mathcal{P}^*}^*$, we have that $X_b^* = X^* \cap T_b^*$. Hence, $X \cap T_b^* = X^* \cap T_b^*$ and (X, F) is a solution for $(\mathcal{I}_b, \mathcal{P})$ which gives $X' \cap T_b = X_b$.

Now let us look at two vertices $u, v \in T \cup (T_b \setminus X_b)$. We need to show that u and v lie in the same connected component of $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$ if and only if they lie in the same connected component of $G_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$. This will conclude the proof of the lemma.

Let $u, v \in T \cup (T_b \setminus X_b)$ lie in the same connected component of $G_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$. Let $D := T \cup (T_b \setminus X_b) \cup Z_W$. Let P be a path connecting u and v in $G_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$ and let $u = v_0, v_1, v_2, \dots, v_r = v$ be the sequence of vertices that lie on P in that order such that $v_i \in D$ for all $i \in \{0, 1, \dots, r\}$. Since $X \setminus V^* = X' \setminus V^*$ and $X \cap D = X' \cap D = X_b \cup X_b^*$, we have $v_i \notin X'$ for all $i \in \{0, 1, 2, \dots, r\}$. Now, to finish this direction of the proof, we need to show that the vertices v_i and v_{i+1} lie in same connected component of $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$ for all $i \in \{0, 1, \dots, r-1\}$.

Let P_i be the subpath of P between v_i and v_{i+1} . As $Z_W \subseteq D$, P_i is either a path in $G'_{\mathcal{P}}$ or a path in $G_{\mathcal{P}}[W]$. For the first case, we know that $X \setminus V^* = X' \setminus V^*$ and also we have that $F \setminus F^* = F' \setminus F^*$, and hence the path is also present on $G'_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$ and hence in $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$. In the second case, we have that $(v_i, v_{i+1}) \in \mathcal{R}_b^*$. As $(X' \cap W, F' \cap E[(G[W])]) = (X^*, F^*)$ is a solution to $(\mathcal{I}_b^*, \mathcal{P}^*)$, we infer that v_i and v_{i+1} are connected via a path P_i^* in $G_{\mathcal{P}^*} - (X' \cap W, F' \cap E[(G[W])])$. Hence, u and v are connected in $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$ if the path P^* does not use any edge $w_1 w_2 \in E_b^*$. By definition of E_b^* , $w_1 w_2 \in E_b^*$ if and only if w_1 and w_2 lie in the same connected component of $G'_{\mathcal{P}} - (X'', F'')$. But we know that $X'' = X \setminus V^* = X' \setminus V^*$ and $F'' = F \setminus E[(G[W])] = F' \setminus E[(G[W])]$ and hence w_1 and w_2 are also connected in $G'_{\mathcal{P}} - \mathcal{X}''_{\mathcal{P}}$ and hence in $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$. This completes one direction of the proof.

For the other direction, the proof is completely symmetrical and we omit the details. This completes the proof of the lemma. \square

Now we describe the recursive step of the algorithm.

Step 4. Assume we are given a B-MMCU instance $\mathcal{I}_b = (G, T, \mathcal{R}, k, l, T_b)$. Invoke first the algorithm of Lemma 2 in a search for $(q, k+l)$ -good node separation (with $V^\infty = T$). If it returns a good node separation (Z, V_1, V_2) , let $j \in \{1, 2\}$ be such that $|V_j \cap T_b| \leq k+l$ and denote $Z^* = Z$, $V^* = V_j$. Otherwise, if it returns that no such good node separation exists in G , invoke the algorithm of Lemma 3 in a search for $(q, k+l)$ -flower separation w.r.t. T_b (with $V^\infty = T$ again). If it returns that no such flower separation exists in G , pass the instance \mathcal{I}_b to the next step. Otherwise, if it returns a flower separation $(Z, (V_i)_{i=1}^l)$, denote $Z^* = Z$ and $V^* = \bigcup_{i=1}^l V_i$.

In the case we have obtained Z^* and V^* (either from Lemma 2 or Lemma 3), invoke the algorithm recursively for the B-MMCU instance \mathcal{I}_b^* defined as in the statement of Lemma 13

for separator Z^* and set V^* , obtaining an output $(\text{sol}_{\mathcal{P}^*}^*)\mathcal{P}^* \in \mathbb{P}(\mathcal{I}_b^*)$. Compute the set $U(\mathcal{I}_b^*)$. Bypass (in an arbitrary order) all vertices of $V^* \setminus (T \cup U(\mathcal{I}_b^*))$. Recall that $T_b^* \subseteq U(\mathcal{I}_b^*)$, so no border terminal gets bypassed. After all vertices of $V^* \setminus U(\mathcal{I}_b^*)$ are bypassed, perform the following operations on terminals of $V^* \cap T$:

1. As long as there exist two different $u, v \in V^* \cap T$ such that $(u, v) \notin \mathcal{R}$, and $uv \in E(G)$, then delete the edge uv and decrease l by 1; if l becomes negative by this operation, return \perp for all $\mathcal{P} \in \mathbb{P}(\mathcal{I}_b)$.
2. As long as there exist two different $u, v \in V^* \cap T$ such that $(u, v) \in \mathcal{R}$ and either $uv \in E(G)$ or $|N_G(u) \cap N_G(v)| > k + l$, identify u and v .
3. If the above two rules are not applicable, then, as long as there exist three pairwise distinct terminals $u_1, u_2, \dots, u_t \in T$ of the same equivalence class of \mathcal{R} that have the same neighborhood and $t > l + 2$, delete u_i for $i > l + 2$ from the graph (and delete all pairs containing u_i from \mathcal{R}).

Let \mathcal{I}_b' be the outcome instance.

Finally, restart this step on the new instance \mathcal{I}_b' and obtain a family of solutions $(\text{sol}_{\mathcal{P}})_{\mathcal{P} \in \mathbb{P}(\mathcal{I}_b')}$ and return this family as an output to the instance \mathcal{I}_b .

We first verify that the application of Lemma 13 is justified in Step 4. By definitions of good node separations and flower separations and by choice of V^* , we have that $|V^* \cap T_b| \leq k + l$ and $G[V^* \cup N_G(V^*)]$ is connected. Also, the recursive calls are applied to strictly smaller graphs because in good node separation, V_2 is deleted in the recursive call while in the other case, by definition of flower separation, we have that $Z \cup \bigcup_{i=1}^l V_i$ is a proper subset of $V(G)$.

After the bypassing operations, we have that V^* contains at most q vertices that are not terminals (at most $k + l$ border terminals and at most $q - (k + l)$ vertices which are neither terminals nor border terminals). Let us now bound the number of terminal vertices once Step 4 is applied. Note that, after Step 4 is applied, for any $v \in T \cap V^*$, we have $N_G(v) \subseteq (V^* \setminus T) \cup Z$ and $|(V^* \setminus T) \cup Z| \leq (q + k + l)$. Due to the first and second rule in Step 4, for any set $A \subseteq (V^* \setminus T) \cup Z$ of size $k + l + 1$, at most one terminal of $T \cap V^*$ is adjacent to all vertices of A . Due to the third rule in Step 4, for any set $B \subseteq (V^* \setminus T) \cup Z$ of size at most $k + l$ and for each equivalence class of \mathcal{R} , there are at most $l + 2$ terminals of this equivalence class with neighborhood exactly B . Let $q' := |T \cap V^*|$, then we have the following.

$$q' \leq (q + k + l)^{k+l+1} + (l + 2)(k + l)(k + l + 1) \sum_{i=1}^{k+l} (q + k + l)^i = 2^{\mathcal{O}((k+l)^2 \log(k+l))}$$

Lemma 14. Assume that we are given a B-MMCU instance $\mathcal{I}_b = (G, T, \mathcal{R}, k, l, T_b)$ on which Step 4 is applied, and let \mathcal{I}_b' be an instance after Step 4 is applied. Then any correct output to the instance \mathcal{I}_b' is a correct output to the instance \mathcal{I}_b as well. Moreover, if Step 4 outputs \perp for all $\mathcal{P} \in \mathbb{P}(\mathcal{I}_b')$, then this is a correct output to \mathcal{I}_b .

Proof. We first note that by Lemma 13, for all $\mathcal{P} \in \mathbb{P}(\mathcal{I}_b)$, for all the vertices $v \notin U(\mathcal{I}_b^*)$, there exists a minimal solution to $(\mathcal{I}_b, \mathcal{P})$ that does not affect v , hence by Lemma 8, the bypassing operation is justified. The second and third rules are justified by Lemma 10 and Lemma 11 respectively. The first rule is justified by Lemma 9, and if application of this rule makes l negative then for any $\mathcal{P} \in \mathbb{P}(\mathcal{I}_b)$, there is no solution to $(\mathcal{I}_b, \mathcal{P})$. \square

Now we do a running time analysis for Step 4. The applications of Lemma 2 and Lemma 3 take time $\mathcal{O}(2^{\mathcal{O}(\min(q, k+l) \log(q+k+l))} n^3 \log n) = \mathcal{O}(2^{\mathcal{O}((k+l)^2 \log(k+l))} n^3 \log n)$. Let $n' = |V^*|$; the

recursive step is applied to a graph with at most $n' + k + l$ vertices and, after bypassing, there are at most $\min(n - 1, n - n' + q + q')$ vertices left. Moreover, each bypassing operation takes $\mathcal{O}(n^2)$ time, the computation of $U(\mathcal{I}_b^*)$ takes $\mathcal{O}(2^{\mathcal{O}((k+l)\log(k+l))}n)$ time. Each application of Lemma 9 takes $\mathcal{O}(n^2)$ time and it can be applied at most l times. So all applications of Lemma 9 take $\mathcal{O}(ln^2)$ time. Application of Lemma 10 takes $\mathcal{O}((k + l)n^2)$ time per operation, which can be implemented by having a counter for each pair of terminals and increasing those counters accordingly by considering every pair of terminals of $N_G(x)$, for each $x \in V(G) \setminus T$. Since when a counter reaches value $k + l + 1$ for vertices u, v , we know that $|N_G(u) \cap N_G(v)| > k + l$, the total time consumed is bounded by $\mathcal{O}((k + l)n^2)$. Application of Lemma 11 takes $\mathcal{O}(n^2 \log n)$ time per operation, since we can sort terminals from one equivalence class according to their sets of neighbours. Since both Lemma 10 and Lemma 11 decrease the number of terminals by at least 1, when applied, they can be applied at most n times. Hence, all applications of Lemma 10 and Lemma 11 takes total $\mathcal{O}(n^3(k + l + \log n))$ time. We also note that values of k and l do not change during the recursive calls. So, we get the following recurrence relation for the running time of Step 4 as a function of vertices of the graph G .

$$T(n) \leq \max_{q+1 \leq n' \leq n-q-1} (\mathcal{O}(2^{\mathcal{O}((k+l)^2 \log(k+l))}n^3 \log n) + T(n' + k + l) + T(\min(n - 1, n - n' + q + q')))$$

Solving the recurrence gives $T(n) = \mathcal{O}(2^{\mathcal{O}((k+l)^2 \log(k+l))}n^4 \log n)$ in the worst case, which is the desired upper bound for the running time of the algorithm.

4.4 High Connectivity phase

In this section we describe the high connectivity phase for the algorithm. Assume we have a B-MMCU instance $\mathcal{I}_b = (G, T, \mathcal{R}, k', l', V^\infty, T_b)$ where Step 4 is not applicable. Let us fix $\mathcal{P} = (X_b, E_b, \mathcal{R}_b, k, l) \in \mathbb{P}(\mathcal{I}_b)$. We iterate through all possible values of \mathcal{P} and try to find a minimal solution to $(\mathcal{I}_b, \mathcal{P})$. Since $|\mathbb{P}(\mathcal{I}_b)| = 2^{\mathcal{O}((k+l)\log(k+l))}$ it results in a factor of $2^{\mathcal{O}((k+l)\log(k+l))}$ in the running time. For a graph G , by $\mathcal{L}(G)$ we denote the set $V(G) \cup E(G)$. Similarly, for a tuple $\mathcal{X} = (X, F)$, by $\mathcal{L}(\mathcal{X})$ we denote the set $X \cup F$. Since we have assumed that Step 4 is not applicable, for any $\mathcal{X} = (X, F)$ where $X \subseteq V(G) \setminus T$, $F \subseteq E(G)$, $|X| \leq k$ and $|F| \leq l$, Lemma 4 implies that the graph $G - \mathcal{X}$ contains at most $t := (2q + 2)(2(k + l) - 1) + 2(k + l) + 1$ connected components containing a non-terminal, out of which at most one can contain more than q vertices outside T . Let us denote its vertex set by $\text{big}(\mathcal{X})$ (observe that this can possibly be the empty set, in case such a component does not exist). We once again need to use lemmas 9-11 to bound number of terminals.

Step 5. Apply Lemma 9, Lemma 10 and Lemma 11 exhaustively on the set T of terminals in the graph (as done in rules 1-3 of Step 4, but doing it for all of T instead of just $T \cap V^*$). Then remove all but $l + 1$ copies of edges having multiplicity more than $l + 1$.

The running time analysis of applying lemmas 9-11 in this step is exactly the same as the one done in Step 4 and takes $\mathcal{O}(n^3(k + l + \log n))$ time. Deletion of copies of parallel edges is justified by Lemma 12 and takes time at most $\mathcal{O}(mn)$. Hence, the total time taken by this step is $\mathcal{O}(n^3(k + l + \log n))$. Now we define the notion of interrogating a solution, which will help us in highlighting the solution.

Definition 5. Let $\mathcal{Z} = (Z, F')$ where $Z \subseteq V(G) \setminus T$, $F' \subseteq E(G)$, $|Z| \leq k$ and $|F'| \leq l$ and let $S \subseteq \mathcal{L}(G) \setminus T$. We say that S interrogates \mathcal{Z} if the following holds:

- $S \cap \mathcal{L}(\mathcal{Z}) = \emptyset$;

- for any connected component C of $G - \mathcal{Z}$ with at most q vertices outside T , all vertices and edges of C belong to $S \cup T$.

Lemma 15. Let $q'' = (qt + k + l)^{k+l+1} + (l+2)(k+l)(k+l+1) \sum_{i=1}^{k+l} (qt + k + l)^i$. Let \mathcal{F} be a family obtained by the algorithm of Lemma 4 for universe $U = \mathcal{L}(G) \setminus T$ and constants $a = q'' + qt + \binom{q''+qt}{2}$ and $b = k + l$. Then, for any $\mathcal{Z} = (Z, F')$ where $Z \subseteq V(G) \setminus T$, $F' \subseteq E(G)$, $|Z| \leq k$ and $|F'| \leq l$, there exists a set $S \in \mathcal{F}$ that interrogates \mathcal{Z} .

Proof. Let $\mathcal{Z} = (Z, F')$ where $Z \subseteq V(G) \setminus T$, $F' \subseteq E(G)$, $|Z| \leq k$ and $|F'| \leq l$. Let A be the union of vertex sets of all connected components of $G - \mathcal{Z}$ that have at most q vertices outside T and let B be the set of edges of these connected components. By Lemma 4, $|A \setminus T| \leq qt$. Also, since we have applied Step 5 exhaustively, we have that no two terminals are adjacent in the graph. Let $C = \{u \mid uv \in E(G) \cap F', v \in A \cap T\}$. Clearly, $C \cap T = \emptyset$, $|C| \leq l$ and for any $v \in A \cap T$, $N(v) \subseteq (A \setminus T) \cup Z \cup C$ which gives $|N(A \cap T)| \leq |(A \setminus T) \cup Z \cup C| \leq qt + k + l$. Now doing the same analysis as in Step 4 gives us $|A \cap T| \leq q''$. Since all the edges in the graph have multiplicity at most $l + 1$, we have that $|A \cup B| \leq q'' + qt + (l+1) \binom{q''+qt}{2}$ and $|\mathcal{L}(\mathcal{Z})| \leq k + l$. So, by Lemma ??, there is a set $S \in \mathcal{F}$ that is disjoint with $\mathcal{L}(\mathcal{Z})$ and contains $(A \setminus T) \cup B$. By construction of A and B , S interrogates \mathcal{Z} . This completes the proof of the lemma. \square

Step 6. Compute the family \mathcal{F} from Lemma 15 and branch into $|\mathcal{F}|$ subcases, indexed by sets $S \in \mathcal{F}$. In a branch S we seek for a minimal solution $\mathcal{X}_{\mathcal{P}}$ to $(\mathcal{I}_b, \mathcal{P})$, which is interrogated by S .

Note that since we have $q'' = 2^{\mathcal{O}((k+l)^2 \log(k+l))}$ and $q, t = 2^{\mathcal{O}((k+l) \log(k+l))}$, the family \mathcal{F} of Lemma 1 is of size $\mathcal{O}(2^{\mathcal{O}((k+l)^3 \log(k+l))} \log n)$ and can be computed in $\mathcal{O}(2^{\mathcal{O}((k+l)^3 \log(k+l))} n \log n)$ time. The correctness of Step 6 is obvious from Lemma 15. As discussed, it can be applied in $\mathcal{O}(2^{\mathcal{O}((k+l)^3 \log(k+l))} n \log n)$ time and gives rise to $\mathcal{O}(2^{\mathcal{O}((k+l)^3 \log(k+l))} \log n)$ subcases.

Lemma 16. Let $\mathcal{X}_{\mathcal{P}} = (X, F)$ be a solution to $(\mathcal{I}_b, \mathcal{P})$ interrogated by S . Then there exists a set $T^{\text{big}} \subseteq T \cup (T_b \setminus X_b)$ that is empty or contains all vertices of exactly one equivalence class of \mathcal{R}_b , such that $X \subseteq (X_b \cup N_G(S(T^{\text{big}})))$ and $F = A_{G,X}(S(T^{\text{big}}))$, where $S(T^{\text{big}})$ is the union of vertex sets of all connected components of $G(S \cup T \cup (T_b \setminus X_b))$ that contain a vertex of $(T \cup (T_b \setminus X_b)) \setminus T^{\text{big}}$ and $A_{G,X}(S(T^{\text{big}}))$ is set of edges in G which have at least one end point in $S(T^{\text{big}})$ but do not belong to any of the connected components of $G[S(T^{\text{big}})]$ and are not incident on X .

Proof. Let $\mathcal{X}_{\mathcal{P}} = (X, F)$. Consider the graph $G_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$ and let $\text{big}_{\mathcal{P}}(\mathcal{X}_{\mathcal{P}})$ be the vertex set of the connected component of $G_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$ that contains $\text{big}(\mathcal{X}_{\mathcal{P}})$ (recall that $G_{\mathcal{P}}$ is the graph G with additional edges E_b ; thus $\text{big}_{\mathcal{P}}(\mathcal{X}_{\mathcal{P}})$ may be significantly larger than $\text{big}(\mathcal{X}_{\mathcal{P}})$). If $\text{big}(\mathcal{X}_{\mathcal{P}}) = \emptyset$, then we take $\text{big}_{\mathcal{P}}(\mathcal{X}_{\mathcal{P}})$ to be the empty set as well. As $\mathcal{X}_{\mathcal{P}}$ is a solution to $(\mathcal{I}_b, \mathcal{P})$, we have $X \cap T_b = X_b$. Define $T^{\text{big}} = (T \cup (T_b \setminus X_b)) \cap \text{big}_{\mathcal{P}}(\mathcal{X}_{\mathcal{P}})$; as $\mathcal{X}_{\mathcal{P}}$ is a solution to $(\mathcal{I}_b, \mathcal{P})$, T^{big} is empty or contains vertices of exactly one equivalence class of \mathcal{R}_b .

Now let C be the vertex set of a connected component of $G - \mathcal{X}_{\mathcal{P}}$ that contains a vertex $v \in (T \cup (T_b \setminus X_b)) \setminus T^{\text{big}}$. Clearly, $v \notin \text{big}_{\mathcal{P}}(\mathcal{X}_{\mathcal{P}})$. As S interrogates $\mathcal{X}_{\mathcal{P}}$, $\text{big}_{\mathcal{P}}(\mathcal{X}_{\mathcal{P}})$ contains $\text{big}(\mathcal{X}_{\mathcal{P}})$ and $X \cap (T \cup T_b) = X_b \subseteq T_b$, we infer that C is the vertex set of a connected component of $G(S \cup T \cup (T_b \setminus X_b))$ as well. As $v \in C$, C is a connected component of $G[S(T^{\text{big}})]$. Let $X' = X_b \cup (X \cap N_G(S(T^{\text{big}})))$ and let $F' = A_{G,X'}(S(T^{\text{big}})) \cap F$. We claim that $\mathcal{X}' = (X', F')$ is a solution to $(\mathcal{I}_b, \mathcal{P})$.

As $X_b \subseteq X$, we have that $X' \subseteq X$ and hence $|X'| \leq k$. Also, since $F' \subseteq F$, we have that $|F'| \leq l$. Moreover, as $X' \subseteq X$, $F' \subseteq F$ and $\mathcal{X}_{\mathcal{P}} = (X, F)$ is a solution to $(\mathcal{I}_b, \mathcal{P})$, if $(u, v) \in \mathcal{R}_b$ then u and v are in the same connected component of $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$. We now show that for any $(u, v) \notin \mathcal{R}_b$ the vertices u and v are in different connected components of $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$. For the sake of contradiction, let $u, v \in T \cup (T_b \setminus X_b)$ such that $(u, v) \notin \mathcal{R}_b$, u and v lie in the same

connected component of $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$ and distance between u and v is minimum possible. Let P be the shortest path between u and v in $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$.

As $\mathcal{X}_{\mathcal{P}} = (X, F)$ is a solution to $(\mathcal{I}_b, \mathcal{P})$, u and v lie in different connected components of $G_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$. Without loss of generality, let $v \notin T^{\text{big}}$ and let C be the connected component of $G - \mathcal{X}_{\mathcal{P}}$ that contains v . Since $(u, v) \notin \mathcal{R}_b$ and $G - \mathcal{X}_{\mathcal{P}}$ is a subgraph of $G_{\mathcal{P}} - \mathcal{X}_{\mathcal{P}}$, we have that $u \notin C$. Since for every edge xy in G such that $x \in C$ and $y \notin C$, we have that either $y \in X'$ or $xy \in F'$, the path P contains an edge $v_1u_1 \in E_b$ such that $v_1 \in C$ and $u_1 \notin C$. So in that case we have that $v_1, u_1 \in T_b$ and hence $(v_1, u_1) \in \mathcal{R}_b$. Also, since $v_1 \in C$, $(v, v_1) \in \mathcal{R}_b$ and since \mathcal{R}_b is an equivalence relation, we have that $(v, u_1) \in \mathcal{R}_b$. But by our assumption, $(u, v) \notin \mathcal{R}_b$, which gives that $(u_1, u) \notin \mathcal{R}_b$. Hence $u_1, u \in T \cup (T_b \setminus X_b)$ such that $(u_1, u) \notin \mathcal{R}_b$ and they are connected by a proper subpath of P in $G_{\mathcal{P}} - \mathcal{X}'_{\mathcal{P}}$, which contradicts our choice of u, v and P . This completes the proof of the lemma. \square

Now we are ready to give the final step of the algorithm. The correctness of the step follows from Lemma 16 and the fact that if S interrogates a solution \mathcal{X} to $(\mathcal{I}_b, \mathcal{P})$, then $|N_G(S(T^{\text{big}}))| \leq k + l$.

Step 7. For each branch, where S is the corresponding guess, we do the following. For each set T^{big} that is empty or contains all vertices of one equivalence class of \mathcal{R}_b , if $|N_G(S(T^{\text{big}}))| \leq k + l$, then for each $X \subseteq X_b \cup N_G(S(T^{\text{big}}))$ such that $|X| \leq k$, and $F = A_{G,X}(S(T^{\text{big}}))$, check whether (X, F) is a solution to $(\mathcal{I}_b, \mathcal{P})$ interrogated by S . For each \mathcal{P} , output a minimal solution to $(\mathcal{I}_b, \mathcal{P})$ that is interrogated by S . Output \perp if no solution is found for any choice of S, T^{big} and X .

Note that \mathcal{R} has at most $(k + l)(k + l + 1)$ equivalence classes. As $|T_b| \leq 2(k + l)$, we have $\mathcal{R}_b \leq (k + l)(k + l + 3)$, and hence there are at most $(k + l)(k + l + 3) + 1$ choices of the set T^{big} . For each T^{big} , computing $N_G(S(T^{\text{big}}))$ and checking whether $|N_G(S(T^{\text{big}}))| \leq k + l$ takes $\mathcal{O}(n^2)$ time. Since $X_b \leq k$, there are at most $(k + 1)(2k + l)^k$ choices for X , and then computing $F = A_{G,X}(S(T^{\text{big}}))$ and checking whether (X, F) is a solution to $(\mathcal{I}_b, \mathcal{P})$ interrogated by S take $\mathcal{O}(n^2)$ time each. Finally, checking whether the solution is minimal or not and computing a minimal solution takes additional $\mathcal{O}((k + l)n^2)$ time. Therefore Step 7 takes $\mathcal{O}(2^{\mathcal{O}((k+l)^3 \log(k+l))} n^2 \log n)$ time for all subcases.

This finishes the description of fixed-parameter algorithm for MMCU and we get the following theorem.

Theorem 3. MMCU can be solved in $\mathcal{O}(2^{\mathcal{O}((k+l)^3 \log(k+l))} n^4 \log n)$ time.

References

- [1] A. Agarwal, N. Alon, and M. Charikar. Improved approximation for directed cut problems. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 671–680, 2007.
- [2] N. Apollonio and B. Simeone. The maximum vertex coverage problem on bipartite graphs. *Discrete Applied Mathematics*, 165:37–48, 2014.
- [3] N. Bousquet, J. Daligault, and S. Thomassé. Multicut is FPT. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 459–468, 2011.
- [4] J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.

- [5] J. Chen, Y. Liu, S. Lu, B. O’Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.
- [6] R. H. Chitnis, M. Cygan, M. Hajiaghayi, M. Pilipczuk, and M. Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 460–469, 2012.
- [7] R. H. Chitnis, M. Hajiaghayi, and D. Marx. Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. *SIAM J. Comput.*, 42(4):1674–1696, 2013.
- [8] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- [9] G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.
- [10] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.
- [11] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. Comput.*, 25(2):235–251, 1996.
- [12] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.
- [13] G. Joret and A. Vetta. Reducing the rank of a matroid. *CoRR*, abs/1211.4853, 2012.
- [14] D. R. Karger, P. N. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Math. Oper. Res.*, 29(3):436–461, 2004.
- [15] K. Kawarabayashi and M. Thorup. The minimum k-way cut of bounded size is fixed-parameter tractable. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 160–169, 2011.
- [16] S. Kratsch, M. Pilipczuk, M. Pilipczuk, and M. Wahlström. Fixed-parameter tractability of multicut in directed acyclic graphs. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, pages 581–593, 2012.
- [17] D. Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.
- [18] D. Marx, B. O’Sullivan, and I. Razgon. Finding small separators in linear time via treewidth reduction. *ACM Transactions on Algorithms*, 9(4):30, 2013.
- [19] D. Marx and I. Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. *SIAM J. Comput.*, 43(2):355–388, 2014.
- [20] R. Niedermeier. *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*. Oxford University Press, USA, March 2006.

- [21] R. Ravi and A. S. II. Approximating k-cuts via network strength. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA.*, pages 621–622, 2002.
- [22] B. A. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.