

Multiway Cuts with a Choice of Representatives^{*}

Kristóf Bérczi^{a,b,1,2}, Tamás Király^{a,b,2}, Daniel P. Szabo^{b,2,*}

^aHUN-REN-ELTE Egerváry Research Group on Combinatorial Optimization,

^bDepartment of Operations Research, Eötvös Loránd University, Budapest, Hungary

Abstract

In the MULTIWAY CUT problem, we are given an undirected graph with non-negative edge weights and a subset of k terminals. The goal is to find a set of edges of minimum total weight whose removal disconnects each terminal from the rest. The problem is APX-hard for $k \geq 3$, and extensive research has focused on closing the gap between the best known upper and lower bounds for approximability and inapproximability.

In this paper, we study several generalizations of MULTIWAY CUT where the terminals can be chosen as some *representatives* from *sets of candidates* T_1, \dots, T_q . In this setting, one may choose these representatives so that the minimum-weight cut separating these sets *via their representatives* is as small as possible. We distinguish different cases based on (A) whether the representative of a candidate set must be separated from the other candidate sets entirely or only from the representatives, and (B) whether there is a single representative for each candidate set or if the choice of representatives is independent for each pair of candidate sets.

For fixed q , we provide approximation algorithms for each of these problems that match the best known approximation guarantee for MULTIWAY CUT. Our main technical contribution is a new extension of the CKR re-

*A conference version of the paper [1] appeared at the 49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024), as well as on Arxiv [2].

*Corresponding Author

Email addresses: kristof.berczi@ttk.elte.hu (Kristóf Bérczi),
tamas.kiraly@ttk.elte.hu (Tamás Király), dszabo2@wisc.edu (Daniel P. Szabo)

¹The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021.

²This research has been supported by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 and ADVANCED 150556 funding schemes.

laxation that preserves these guarantees. For general q , we show $o(\log q)$ -inapproximability for all cases where the choice of representatives may depend on the pair of candidate sets, as well as for the case where the goal is to separate a fixed node from a single representative from each candidate set. As a positive result, we present a 2-approximation algorithm for the case where we need to choose a single representative from each candidate set. This setting generalizes k -CUT, and our algorithm addresses it by relating the tree case to optimization over a gammoid.

Keywords: Representative Cuts, Approximation Algorithms, Multiway Cut, CKR Relaxation, Steiner Multicut

1. Introduction

For an undirected graph $G = (V, E)$ with weight function $w: E \rightarrow \mathbb{R}_+$, the MULTIWAY CUT problem asks for a minimum-weight cut $C \subseteq E$ separating any pair of terminals in a given terminal set $S = \{s_1, \dots, s_k\}$. As cuts can be identified with partitions of the node set, this is equivalent to finding a node coloring of G with k colors such that terminal s_i is colored with color i for $i \in [k]$, and we seek to minimize the total weight of dichromatic edges.

Previous work. Dahlhaus et al. [3] showed that MULTIWAY CUT is NP-hard even for $k = 3$, and provided a very simple combinatorial $(2 - 2/k)$ -approximation that works as follows. For each s_i , determine a minimum-weight cut $C_i \subseteq E$ that separates s_i from s_j for $j \neq i$ – such a cut is called an *isolating* cut of s_i – and then take the union of the $k - 1$ smallest ones among the k cuts thus obtained. In an optimal multiway cut, the boundary of the component containing s_i is a cut isolating s_i , hence its weight is at least as large as that of C_i . Summing up these inequalities for all but the largest isolating cuts, since this counts each edge at most twice except for the boundary of the largest one, leads to a $(2 - 2/k)$ -approximation.

Since the pioneering work of Dahlhaus et al., MULTIWAY CUT has been a central problem in combinatorial optimization. The best known approximability as well as inapproximability bounds are based on a geometric relaxation called the *CKR relaxation*, introduced by Călinescu, Karloff and Rabani [4]. The current best approximation algorithm is due to Sharma and Vondrák [5] with an approximation factor of 1.2965, while the best known lower bound (assuming the Unique Games Conjecture) is slightly above 1.2 [6].

Various generalizations of MULTIWAY CUT have been introduced. In the MULTICUT problem, we are given an undirected graph with non-negative edge weights, together with a demand graph consisting of edges s_1t_1, \dots, s_kt_k , and the goal is to determine a minimum-weight cut whose removal disconnects each s_i from its pair t_i . The UNIFORM METRIC LABELING problem takes as input a list of possible colors for each node in an edge-weighted graph, and asks for a coloring that respects these lists with the minimum total weight of dichromatic edges³; MULTIWAY CUT arises as a special case when the terminals have distinct lists of length 1 and all other nodes can be colored arbitrarily. Kleinberg and Tardos [7] gave a 2-approximation to UNIFORM METRIC LABELING with a tight integrality gap using a geometric relaxation, similar to that of CKR. In the k -CUT problem, we are given only an edge-weighted graph G and a positive integer k , and the goal is to find a minimum-weight cut whose deletion breaks the graph into k components. One can think of this problem as a version of MULTIWAY CUT where the terminals can be chosen freely. The k -CUT problem admits a 2-approximation [9] that is tight [10].

Another generalization of k -CUT and MULTIWAY CUT is STEINER k -CUT [11], which is closely related to our work. In this problem, only a single terminal set is given along with an integer k , and the output is the minimum k -cut in which each component has at least one terminal. The STEINER MULTICUT [12] problem takes as input an undirected graph G and subsets X_1, \dots, X_q of nodes, and asks for a minimum cut such that each X_i is separated into at least 2 components. A generalization of STEINER MULTICUT is the REQUIREMENT CUT problem [13], where requirements r_i are given for each set X_i , and the goal is to find the minimum cut that cuts each X_i into at least r_i components. The current best algorithms for REQUIREMENT CUT are those given in [13, 14], of which we will use the $O(\log k \log q)$ approximation, where $k = |\bigcup_{i=1}^q X_i| \leq n$.

Our results. As mentioned above, k -CUT can be considered as a version of MULTIWAY CUT where the terminals can be chosen freely, and STEINER k -CUT can be seen as a common generalization of these two problems that

³The original definition of [7] differs from the version we define here in that it allows arbitrary assignment weights of each node to a terminal, but as shown in [8], this restricted version is equivalent.

restricts the choice of terminals. We present a novel framework that encompasses these problems, but allows a finer control of the choice of terminals than STEINER k -CUT.

We introduce and study generalizations of MULTIWAY CUT where we are allowed to choose *representatives* from given terminal candidate sets $T_1, \dots, T_q \subseteq V$, and the goal is to find the minimum-weight cut separating these sets *via their representatives*. We offer a comprehensive study of several variants of the problem. These variants are distinguished by (A) whether the representative has to be separated from all candidates of the other candidate sets or only from their representatives, and (B) whether there is a single representative for each candidate set or whether the choice of representative is independent for each pair of candidate sets. In order to make it easier to distinguish these problems, on a high level, we use the following naming convention.

- When the goal is to separate *all* candidates, we use ALL; for example, the ALL-TO-ALL problem requires all nodes of T_i to be separated from all nodes of T_j , for each $i \neq j$.
- When the goal is to choose a *single* representative for each candidate set, we use SINGLE, and we denote the chosen representative of T_i by t_i . For example, the SINGLE-TO-SINGLE problem requires choosing a representative $t_i \in T_i$ for every $i \in [q]$, and finding a cut that separates t_i from t_j for all $i \neq j$. On the other hand, the SINGLE-TO-ALL problem requires the chosen representative $t_i \in T_i$ to be separated from every node of T_j , for all $i \neq j$.
- When only *some* representative of T_i ought to be separated from some part of T_j for each i, j pair, we use SOME, and denote the representative chosen from T_i to be separated from part of T_j by t_i^j . For example, the SOME-TO-SOME problem asks for a minimum-weight subset of edges such that after deleting these edges, for any pair $i \neq j$, there are nodes $t_i^j \in T_i$ and $t_j^i \in T_j$ that are in different components.
- When there is a *fixed* node that needs to be separated from the candidate sets, we use FIXED, and denote the fixed node by s . For example, the FIXED-TO-SINGLE problem asks for a minimum-weight subset of edges such that after deleting these edges, s is separated from at least one element $t_j \in T_j$ for every $j \in [q]$.

These problems are natural generalizations of MULTIWAY CUT that provide various ways to interpolate between problems with fixed terminals like MULTIWAY CUT and problems with freely chosen terminals like k -CUT. For example, MULTIWAY CUT, k -CUT, and STEINER k -CUT arise as special cases of SINGLE-TO-SINGLE when the terminal sets are all singletons, the entire node set, or the given terminal set, respectively. Our framework allows for a more fine-grained generalization than STEINER k -CUT, while still achieving the same positive results. Although some of our problems are equivalent or closely related to those previously studied in the literature, no systematic study of this type of generalization has been done previously. In addition, some of our results require new observations and techniques.

In each problem, we want to minimize over all possible choices of representatives, as well as over all possible subsets of edges. The problem where we need to separate each candidate set from every other, ALL-TO-ALL, is equivalent to MULTIWAY CUT by contracting each candidate set to a single node. The other problems are not directly reducible to MULTIWAY CUT. We denote by $\alpha \approx 1.2965$ the current best approximation factor for MULTIWAY CUT [5]. The different problems, as well as our results, are summarized in Table 1. The main results that require new techniques are indicated in bold in the table, and are discussed in the next subsection.

Problem	Demands	Fixed q	Unbounded q
ALL-TO-ALL	$T_i - T_j$	α -approx	α -approx
SINGLE-TO-SINGLE	$t_i - t_j$	α -approx	Tight 2-approx
SINGLE-TO-ALL	$t_i - T_j$	α-approx	2-approx
FIXED-TO-SINGLE	$s - t_j$	In P	No $o(\log q)$ approx
SOME-TO-SINGLE	$t_i^j - t_j$	α -approx	No $o(\log q)$ approx
SOME-TO-SOME	$t_i^j - t_j^i$	α -approx	$O(\log q \cdot \log n)$ approx [13]
SOME-TO-ALL	$t_i^j - T_j$	α-approx	No $o(\log q)$ approx

Table 1: A summary of our results, where $\alpha \approx 1.2965$ [5] is the current best approximation factor for MULTIWAY CUT. The tightness of 2-approximation assumes SSEH (see Proposition 2), while the other inapproximability results hold assuming P \neq NP. The main results are highlighted in bold.

Techniques.

Approximation when q is part of the input. We give 2-approximations for SINGLE-TO-ALL and SINGLE-TO-SINGLE. For the latter, we first give an exact algorithm on trees, by showing that the feasible solutions have a gammoid structure. This then leads to a 2-approximation for general graphs using the Gomory-Hu tree, which is best possible, since SINGLE-TO-SINGLE generalizes the k -CUT problem which has no $2 - \varepsilon$ approximation by Proposition 2. Also, we show that the SOME-TO-SOME problem is equivalent to STEINER MULTICUT, leading to an $O(\log q \cdot \log n)$ approximation in this case.

Approximation for fixed q . Some of the problems with fixed q are directly reducible to solving a polynomial number of MULTIWAY CUT instances. However, this is not the case for SINGLE-TO-ALL and SOME-TO-ALL. Our α -approximation algorithms for these are obtained by extending the CKR relaxation to a more general problem that we call LIFTED CUT (see Section 4) in such a way that the rounding methods used in [5] still give an α -approximation. LIFTED CUT may have independent interest as a class of metric labeling problems that is broader than MULTIWAY CUT but can still be approximated to the same ratio. We then show that for fixed q , problems SINGLE-TO-ALL and SOME-TO-ALL are reducible to solving polynomially many instances of LIFTED CUT.

Hardness of approximation. We prove hardness of FIXED-TO-SINGLE by reducing from HITTING SET, the hardness of which comes from Proposition 1. We then reduce SOME-TO-ALL, SOME-TO-SINGLE, and SOME-TO-SOME from FIXED-TO-SINGLE to give hardness results for those problems as well.

Structure of the paper. In Section 2, we present the definitions and the main tools used in our algorithms and proofs. Section 3 presents a greedy 2-approximation for the SINGLE-TO-SINGLE problem. Section 4 introduces the LIFTED CUT problem and describes how to extend the α -approximation of [5] to LIFTED CUT. Sections 5 and 6 use this formulation to solve the other variants in the regime where q is fixed. Finally, section 7 presents some hardness results, as well as the equivalence of SOME-TO-SOME to STEINER MULTICUT.

2. Preliminaries

Throughout the paper, we denote the set of non-negative reals by \mathbb{R}_+ , and use $[k] = \{1, \dots, k\}$. Given an undirected graph $G = (V, E)$, the edge going between nodes $u, v \in V$ is denoted by uv . For a weight function $w: E \rightarrow \mathbb{R}_+$ and $C \subseteq E$, we use $w(C) = \sum_{e \in C} w(e)$. The graph obtained by deleting the edges in C is denoted by $G - C$. We denote the set of components of G by $\mathcal{K}(G)$. The boundary of a given subset of nodes $S \subseteq V$ is $\delta(S) = \{uv \in E : u \in S, v \in V \setminus S\}$. We use e^i to denote i th elementary vector. The k -dimensional simplex if defined as $\Delta_k = \{x \in \mathbb{R}_+^k : \sum_{i=1}^k x_i = 1\}$.

In what follows, we give formal definitions of the problems that we consider and briefly summarize the background results that we build upon in our proofs.

Problem Formulation. In this subsection, we provide precise problem formulations and characterizations for the existence of feasible solutions. The reader may skip this subsection if the intuitive definitions in the introduction were clear. In all our problems, the input is an undirected graph $G = (V, E)$, a weight function $w: E \rightarrow \mathbb{R}_+$, and, not necessarily disjoint, candidate sets $T_1, \dots, T_q \subseteq V$. We consider the following variants.

ALL-TO-ALL

Goal: Determine

$$\arg \min_{C \subseteq E} \{w(C) : \forall H \in \mathcal{K}(G - C) \text{ s.t. } H \cap T_i \neq \emptyset, H \cap T_j = \emptyset \forall j \neq i\}.$$

If $T_i \cap T_j \neq \emptyset$ for some distinct $i, j \in [q]$, then ALL-TO-ALL admits no feasible solution. Otherwise, $C = E$ is a feasible solution and the minimum is well defined.

SINGLE-TO-ALL

Goal: Determine

$$\arg \min_{C \subseteq E, t_i \in T_i} \{w(C) : \forall H \in \mathcal{K}(G - C) \text{ s.t. } t_i \in H, H \cap T_j = \emptyset \forall j \neq i\}.$$

If $T_i \subseteq \bigcup_{j \in [q] \setminus \{i\}} T_j$ for some $i \in [q]$, then SINGLE-TO-ALL admits no feasible solution. Otherwise, if we set t_i be any element of $T_i \setminus \bigcup_{j \in [q] \setminus \{i\}} T_j$ for $i \in [q]$, then $C = E$ is a feasible solution.

SINGLE-TO-SINGLE

Goal: Determine

$$\arg \min_{C \subseteq E, t_i \in T_i} \{w(C) : \forall H \in \mathcal{K}(G - C), |\{t_1, \dots, t_q\} \cap H| \leq 1\}.$$

Note that in this case the representatives t_i form a *transversal*, or a system of distinct representatives, for the set system T_1, \dots, T_q . Thus, SINGLE-TO-SINGLE is feasible exactly if such a transversal exists, or in other words, there exists a matching of size q in the bipartite graph corresponding to the set system.

SOME-TO-SINGLE

Goal: Determine

$$\arg \min_{C \subseteq E, t_i \in T_i, t_i^j \in T_i} \{w(C) : \forall H \in \mathcal{K}(G - C) \text{ s.t. } t_i \in H, t_j^i \notin H \ \forall j \neq i\}.$$

Equivalently, we can rephrase the problem as

$$\arg \min_{C \subseteq E, t_i \in T_i} \{w(C) : \forall H \in \mathcal{K}(G - C) \text{ s.t. } t_i \in H, T_j \setminus H \neq \emptyset \ \forall j \neq i\}.$$

If there is a set T_j such that each $v \in T_j$ appears as a singleton set $T_i = \{v\}$, then no representative t_j of T_j can satisfy the above cut demands, and SOME-TO-SINGLE admits no feasible solution. Otherwise, each T_j must have some element t_j that does not appear as a singleton set, which means that each candidate set has some t_i^j that is distinct from t_j . This set of representatives forms a feasible solution with $C = E$.

SOME-TO-ALL

Goal: Determine

$$\arg \min_{C \subseteq E, t_i^j \in T_i} \{w(C) : \forall H \in \mathcal{K}(G - C) \text{ s.t. } t_i^j \in H, H \cap T_j = \emptyset \ \forall j \neq i\}.$$

If $T_i \subseteq T_j$ holds for some $i \neq j$, then no element of T_i can be separated from all of T_j , and there SOME-TO-ALL admits no feasible solution. Otherwise, any choice of representatives where $t_i^j \in T_i \setminus T_j$ for all $i, j \in [q], i \neq j$ gives a feasible solution with $C = E$.

SOME-TO-SOME

Goal: Determine

$$\arg \min_{C \subseteq E, t_i^j \in T_i} \{w(C) : \forall H \in \mathcal{K}(G - C), \{t_i^j, t_j^i\} \not\subseteq H \ \forall i \neq j\}.$$

If two candidate sets T_i, T_j are both singleton sets consisting of the same node, then SOME-TO-SOME admits no feasible solution. Otherwise, for any intersecting pair $i, j \in [q], i \neq j$, at least one of the sets has size ≥ 2 , and we can pick two distinct elements as $t_i^j \in T_i$ and $t_j^i \in T_j$. For each other disjoint pair we can then choose t_j^i, t_i^j arbitrarily to get a feasible solution with $C = E$.

The input of the following problem also contains a fixed node $s \in V$.

FIXED-TO-SINGLE

Goal: Determine

$$\arg \min_{C \subseteq E, t_i \in T_i} \{w(C) : \forall H \in \mathcal{K}(G - C) \text{ s.t. } s \in H, t_j \notin H \forall j\}.$$

If $T_i = \{s\}$ for some index i , then FIXED-TO-SINGLE admits no feasible solution. Otherwise, the choice $C = E$ and $t_j \in T_j \setminus \{s\}$ gives a feasible solution.

The CKR Relaxation and Rounding Methods. For a graph $G = (V, E)$ with edge weights $w: E \rightarrow \mathbb{R}_+$ and terminals $S = \{s_1, \dots, s_k\}$, the CKR relaxation [4] is the linear program LP 1 that assigns to each node a geometric location in the k -dimensional simplex. Note that the ℓ_1 -norm appearing in the objective function can be converted to a linear objective by standard arguments, see e.g. [15].

CKR-LP:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \sum_{uv \in E} w_{uv} \|x^u - x^v\|_1 \\ \text{subject to} \quad & x^u \in \Delta_k \quad u \in V, \\ & x^{s_i} = e^i \quad i \in [k]. \end{aligned}$$

UML-LP:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \sum_{uv \in E} w_{uv} \|x^u - x^v\|_1 \\ \text{subject to} \quad & x^u \in \Delta_q \quad u \in V, \\ & x_i^v = 0 \quad i \notin \ell(v). \end{aligned}$$

LP 1: CKR relaxation of multiway cut.

LP 2: Relaxation for UNIFORM METRIC LABELING.

The original paper of Călinescu, Karloff and Rabani [4] gives a $(3/2 - 1/k)$ -approximation algorithm that works as follows. First, take a threshold $\rho_i \in (0, 1)$ uniformly at random for each dimension $i \in [k]$. Then let σ be chosen uniformly at random from the permutations $(1, 2, \dots, k-2, k-1, k)$ and $(k-1, k-2, \dots, 2, 1, k)$, and assign nodes within a distance $\rho_{\sigma(i)}$ of $x^{s_{\sigma(i)}}$ to the component of $s_{\sigma(i)}$ for $i \in [k-1]$, and assign the remaining nodes to s_k . We call an algorithm that chooses some thresholds ρ_i as well as a permutation σ of the terminals and then assigns the nodes within the threshold $\rho_{\sigma(i)}$ to the terminals in the σ order a *threshold algorithm*.

Algorithm 1 The Kleinberg-Tardos Algorithm for UNIFORM METRIC LABELING.

Input: A graph $G = (V, E)$, weights $w: E \rightarrow \mathbb{R}_+$, labels $\ell: V \rightarrow \mathcal{P}([q])$, and a solution $x \in \mathbb{R}^V$ of UML-LP.

Output: A solution to UNIFORM METRIC LABELING.

- 1: **while** there is $u \in V$ s.t. u is unassigned **do**
 - 2: Pick a label $i \in [q]$ uniformly at random, and a threshold $\rho \sim \text{unif}[0, 1]$.
 - 3: Assign label i to any unassigned $u \in V$ with $x_i^u \geq \rho$.
 - 4: **end while**
-

The analysis of the above linear programming formulation revealed several useful properties of the CKR relaxation. One of these observations is that the edges of the graph may be assumed to be *axis-aligned*. An edge uv is said to be (i, j) -axis-aligned if x^u and x^v differ only in coordinates i and j . Roughly speaking, any edge that is not axis-aligned can be subdivided into several edges that are axis-aligned, forming a piecewise linear path between x^u and x^v . This observation significantly simplifies the analysis of threshold algorithms, as there are at most two thresholds that can cut any axis-aligned edge. Another useful property is *symmetry*. For any threshold algorithm, there is one that achieves the same guarantees by choosing a uniformly random permutation. See [16, Section 2] for a more detailed discussion of these properties.

Another way of rounding the CKR relaxation is provided by the *exponential clocks* algorithm of Buchbinder, Naor and Schwartz [17]. Their approach can be thought of as choosing a uniformly random point in the simplex, and splitting the simplex by axis parallel hyperplanes that meet at this given point. The algorithm gives the same guarantees as the algorithm of Kleinberg and Tardos [7] for UNIFORM METRIC LABELING. This latter problem takes as input a list of possible colors $\ell(v)$ for each node v in a given graph, and asks for a coloring that respects these lists with the minimum total weight of dichromatic edges. Their relaxation LP 2 is similar to the CKR relaxation when there are a total of q colors, but it does not require there to be nodes at every vertex of the simplex. It is shown in [17, Section 6] that Algorithm 1 gives the same guarantees as the exponential clocks algorithm.

The approximation algorithm of Sharma and Vondrák [5] for MULTIWAY CUT uses a carefully designed randomized scheme that selects among four algorithms of the above two types according to a non-uniform distribution, with an analysis showing that this yields an α -approximation, where $\alpha \approx 1.2965$.

Other Relevant Tools. Our hardness of approximation results are based on two different complexity assumptions. The $o(\log q)$ inapproximability results hold assuming $P \neq NP$, based on the hardness of approximating HITTING SET.

Proposition 1 ([18, 19]). *For any fixed $0 < \alpha < 1$, HITTING SET cannot be approximated in polynomial time within a factor of $(1 - \alpha) \ln N$ on inputs of size N , unless $P = NP$.*

The other complexity assumption that we use is the *Small Set Expansion Hypothesis* (SSEH), a core hypothesis for proving hardness of approximation for problems that do not have straightforward proofs assuming the Unique Games Conjecture (UGC). It implies the UGC, and we will use it as evidence against a $(2 - \varepsilon)$ -approximation, for any $\varepsilon > 0$, for k -CUT.

Proposition 2 ([10]). *Assuming SSEH, k -CUT cannot be approximated in polynomial time within a factor of $(2 - \varepsilon)$, for any constant $\varepsilon > 0$.*

From matroid theory, we use the notion of gammoids. A *gammoid* $M = (D, S, T)$ is a matroid defined by a digraph $D = (V, E)$, a set of source nodes $S \subseteq V$, and a set of target nodes $T \subseteq V$. A set $X \subseteq T$ is independent in M if there exist $|X|$ node-disjoint paths from elements of S into X .

Finally, Gomory-Hu (GH) tree is a standard tool in graph cut algorithms. The *GH tree* of a graph $G = (V, E)$ with weight function $w: E \rightarrow \mathbb{R}_+$ is a tree $T = (V, F)$ together with weight function $w_T: F \rightarrow \mathbb{R}_+$ that encodes the minimum-weight $s - t$ cuts for each pair s, t of nodes in the following sense: the minimum w_T -weight of an edge on the $s - t$ path in T is equal to the minimum w -weight of a cut in G separating s and t . Furthermore, the two components of the tree obtained by removing the edge of minimum w_T -weight on the path give the two sides of a minimum w -weight $s - t$ cut in G .

3. Single-to-Single Problem

In this problem, we are looking for a *single* representative from each candidate set together with a minimum multiway cut separating them. Note that when $T_1 = \dots = T_q$, SINGLE-TO-SINGLE corresponds to STEINER k -CUT with $k = q$.

Theorem 3. *For fixed q , there is a polynomial-time α -approximation algorithm for SINGLE-TO-SINGLE.*

Proof. When q is fixed, one can iterate through all the $O(n^q)$ possible choices of representatives, approximate the corresponding MULTIWAY CUT instance, and choose the best one. \square

Algorithm 2 Greedy algorithm for SINGLE-TO-SINGLE on trees.

Input: A tree $G = (V, E)$, weights $w: E \rightarrow \mathbb{R}_+$, and $T_1, \dots, T_q \subseteq V$.

Output: A minimum-weight good cut $C \subseteq E$.

- 1: Set $C \leftarrow \emptyset$.
 - 2: **while** $|C| < q - 1$ **do**
 - 3: Pick an $e \in \arg \min\{w(e): e \notin C, C + e \text{ is good}\}$.
 - 4: $C = C + e$
 - 5: **end while**
-

For general q , it is helpful to first look at the case where G is a tree. We show that in this special case, the problem reduces to finding the minimum cost basis of a gammoid. We call a cut $C \subseteq E$ *good* if $G - C$ has a valid set of representatives, that is, if we can choose $|C| + 1$ representatives that form a partial transversal of the candidate sets, and each component of $G - C$ contains a single representative from this partial transversal. The algorithm is presented as Algorithm 2.

Theorem 4. *Algorithm 2 computes an optimal solution to SINGLE-TO-SINGLE on trees.*

Proof. We prove the statement by showing that the problem is equivalent to optimizing over a gammoid. We construct a directed graph as follows. Let $r \in V$ be an arbitrary root node, and orient the edges of the tree towards r . For a non-root node v , we denote the unique arc leaving v by $e(v)$ and define the cost of v to be $w(e(v))$. Furthermore, for each set T_i , we add a node s_i together with arcs from s_i to the candidates in T_i .

Let D denote the digraph thus obtained, set $S := \{s_1, \dots, s_q\}$ and $T := V$, and consider the gammoid $M = (D, S, T)$. The key observation is the following.

Claim 5. *For a set $Z \subseteq V \setminus \{r\}$, $C = \{e(v): v \in Z\}$ is a good cut if and only if $Z \cup \{r\}$ is independent in M .*

Proof. For the forward direction, assume that $C = \{e(v): v \in Z\}$ forms a good cut. Let $Z = \{v_1, \dots, v_p\}$. Without loss of generality, we may assume that the candidate sets having a valid set of representatives in $G - C$ are T_1, \dots, T_{p+1} , where v_i is in the component of the representative t_i of T_i and r is in the same component as the representative t_{p+1} of T_{p+1} . For $i \in [p]$, the edge $s_i t_i$ and the path $t_i - v_i$ in the tree form an $s_i - v_i$ path; similarly, the edge $s_{p+1} t_{p+1}$ and the path $t_{p+1} - r$ in the tree form an $s_{p+1} - r$ path. Furthermore, these paths are pairwise node-disjoint, since they use different connected components of $G - C$.

For the other direction, assume that $Z \cup \{r\}$ is independent in M , and let $Z = \{v_1, \dots, v_p\}$. Without loss of generality, we may assume that there are pairwise

Algorithm 3 Approximation algorithm for SINGLE-TO-SINGLE on graphs.

Input: A graph $G = (V, E)$, weights $w: E \rightarrow \mathbb{R}_+$, $T_1, \dots, T_q \subseteq V$.

Output: A feasible cut $C \subseteq E$.

- 1: Compute the Gomory-Hu tree H of G .
 - 2: Run Algorithm 2 on H .
 - 3: Return the union of the cuts corresponding to edges found in Step 2.
-

node-disjoint paths from s_i to v_i for $i \in [p]$ together with a path from s_{p+1} to r . Let $t_i \in T_i$ be the first node on the path starting from s_i for $i \in [p+1]$. Then $\{t_1, \dots, t_{p+1}\}$ form a valid system of distinct representatives for the cut C as each of these nodes are in a separate component of $G - C$. \square

By Claim 5, a minimum-weight maximum-sized good cut can be determined using the greedy algorithm. To see this, note that the greedy algorithm on the gammoid defined above proceeds by picking a vertex set, starting with r . Since there is a one-to-one correspondence between edges and vertices other than r , adding a vertex to the solution in the gammoid can be interpreted as adding the corresponding edge to the solution in the original problem. Algorithm 2 performs exactly these steps. \square

Algorithm 2 solves the special case when G is a tree. The classical $(2 - 2/k)$ approximation for MULTIWAY CUT uses 2-way cuts coming from the Gomory-Hu tree, and so does the $(2 - 2/k)$ approximation for k -CUT [20]. We follow a similar approach in Algorithm 3. The algorithm can be interpreted as taking the minimum edges in the GH tree as long as they allow a valid system of representatives.

Theorem 6. *Algorithm 3 computes a $(2 - 2/q)$ approximation to SINGLE-TO-SINGLE on arbitrary graphs.*

Proof. Let OPT be the optimal solution with representatives t_1^*, \dots, t_q^* and components V_1^*, \dots, V_q^* , where V_q^* has the maximum weight boundary $\delta(V_q^*)$. Furthermore, let H be the GH tree of G .

We transform OPT into a solution OPT_{GH} on H , losing at most a factor of $(2 - 2/q)$. This transformation is carried out by iteratively removing minimum-weight edges in $E(H)$ that separate pairs of representatives t_1^*, \dots, t_q^* belonging to the same component of H . Formally, we initialize $H_0 = H$. At step $\ell \geq 1$, if there exist two representatives that lie in the same component of $H_{\ell-1}$, we select a minimum-weight edge $e_\ell \in E(H_{\ell-1})$ that separates a pair t_i^* and t_j^* of representatives in OPT . We then define $f_\ell = t_i^* t_j^*$ and set $H_\ell = H_{\ell-1} - e_\ell$. Repeating this process

yields a sequence of edges e_1, \dots, e_{q-1} and a tree of representative pairs $F = (\{t_1^*, \dots, t_q^*\}, \{f_1, \dots, f_{q-1}\})$.

Note that F is indeed a tree. Suppose, for the sake of contradiction, that this is not the case. Then there exist indices i_1, \dots, i_p such that $t_{i_1}^*, \dots, t_{i_p}^*$ are the vertices of a cycle in this order, and the edge $t_{i_1}^* t_{i_2}^*$ is the first one added to F , say, in step ℓ . But this means that after adding this edge, $t_{i_1}^*$ and $t_{i_2}^*$ lie in different components of H_ℓ , so there must be at least one other edge $t_{i_r}^* t_{i_{r+1}}^*$ in the cycle whose endpoints also belong to different components of H_ℓ . This contradicts the fact that $t_{i_r}^* t_{i_{r+1}}^*$ was later added to F .

Fix a vertex x arbitrarily. For an edge $e \in E(H)$, let $U(e)$ denote the vertices of the connected component of $H - e$ containing x . Direct the edges of F away from t_q^* , and reorder the edges such that f^i is the edge going into t_i^* for $i \in [q-1]$. Let e^i be the edge of the GH tree corresponding to f^i , i.e., the minimum weight edge of the path between the two endpoints of f^i . Then the boundary of each component satisfies $w(\delta(V_i^*)) \geq w(U(e^i))$, as $\delta(V_i^*)$ separates the two representatives in f^i as well, and $U(e^i)$ is the minimum-weight cut between those.

Let the solution OPT_{GH} be $\bigcup_{i \in [q-1]} U(e_i)$, ALG be the cut found by the algorithm, ALG_{GH} be the corresponding edges in the GH tree H , and w_H be the weight function on H . Then

$$\begin{aligned} w(ALG) &\leq w_H(ALG_{GH}) \leq w_H(OPT_{GH}) = \sum_{i=1}^{q-1} w(U(e^i)) \leq \sum_{i=1}^{q-1} w(\delta(V_i^*)) \\ &\leq (1 - 1/q) \sum_{i=1}^q w(\delta(V_i^*)) \leq (2 - 2/q)w(OPT). \end{aligned}$$

This concludes the proof of the theorem. \square

4. Lifted Cuts

The goal of this section is to show that the following restriction of the UNIFORM METRIC LABELING relaxation to a one-dimensional lifting of the CKR relaxation admits an α -approximation to its integer optimum. We define the lifted cut problem LIFTED CUT, which takes as input a graph $G = (V, E)$ with edge-weights $w: E \rightarrow \mathbb{R}_+$, fixed terminals $S = \{s_1, \dots, s_q\} \subseteq V$, and a list of possible colors for each node $\ell: V \rightarrow \mathcal{P}[q+1]$, the power set of $[q+1]$, satisfying the following two conditions:

1. $\ell(s_i) = \{i\}$ for $i \in [q]$,
2. $q+1 \in \ell(v)$ for all $v \in V \setminus S$.

The goal is then to assign a color to each node from its list such that the total weight of dichromatic edges is minimized. We consider the following linear programming relaxation of LIFTED CUT.

LIFT-LP:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \sum_{uv \in E} w_{uv} \|x^u - x^v\|_1 \\ \text{subject to} \quad & x^u \in \Delta_{q+1} \quad u \in V, \\ & x_i^u = 0 \quad i \notin \ell(u). \end{aligned}$$

LP 3: Relaxation of LIFTED CUT.

Condition 1 ensures that the set S indeed defines terminals that vertices of the simplex are assigned to, as in MULTIWAY CUT, but Condition 2 offers a relaxation, allowing a vertex of the simplex not to be assigned to any terminal. This condition gives an additional dimension to the simplex, while still preserving the approximation guarantees given by the best currently known rounding algorithms for CKR.

Theorem 7. *There is a polynomial-time α -approximation algorithm for LIFTED CUT.*

Proof. The high-level idea of the proof is to show that the rounding scheme of [5], when applied to LIFT-LP using Algorithm 1 in place of the exponential clocks algorithm, and with the modification that only the first q coordinates are permuted in the threshold algorithms while coordinate $q + 1$ is always left last, gives an α -approximation to LIFTED CUT.

First, we have to argue that the threshold algorithms give feasible solutions to LIFTED CUT (for Algorithm 1, this follows since LIFTED CUT is a metric labeling problem). In all algorithms, s_i is assigned to the i th component, since x^{s_i} is the i th vertex of the simplex. For other nodes $v \in V \setminus S$, $x_i^v = 0$ guarantees that v is not assigned to the i th component if $i \notin \ell(v)$. Here, we use the fact that the $(q + 1)$ st component is the only one for which there is no threshold. Although it is possible that $x_{q+1}^v = 0$ and v is still assigned to the $(q + 1)$ st component, this is not a problem, because $q + 1 \in \ell(v)$ by definition.

To prove that we get an α -approximation, we need to show that the relevant bounds that are used in the analysis of the four algorithms mentioned in Sharma-Vondrák [5] carry through to this modified LIFT-LP. We give a sketch here, but the details are written out more carefully in Appendix A. We consider the two types of algorithms (i.e. threshold and exponential clocks) separately.

It was observed in [17] that the exponential clocks algorithm can be replaced by the 2-approximation for the UNIFORM METRIC LABELING problem of Kleinberg-Tardos [7]. Since LIFT-LP corresponds to a UNIFORM METRIC LABELING problem, the bound in [17, Lemma 3] remains valid in our case. Since this is the key bound for the exponential clocks algorithm used in the analysis, we can conclude that Algorithm 1 for LIFT-LP gives the same guarantees.

The other algorithms we need to argue the validity of are the threshold algorithms. These assume that there is a node at every vertex of the simplex, which is not necessarily true for the LIFT-LP as no variable needs to be at e^{q+1} . We can, however, use the fact that there is only *one* such vertex, and change the order of the terminals so that this vertex is cut last. We can then use the analysis in [5] and [17] of the CKR relaxation for $k = q + 1$, as sketched below.

The threshold algorithms choose a permutation of the nodes to achieve some symmetry, which is necessary for *only* the first $k - 1$ terminals. The last terminal, which is just assigned the remaining nodes, *does not have its own threshold*. In each of the Single Threshold, Descending Threshold and Independent Threshold algorithms of [5], they prove results for the first two indices, and then argue that these hold for any pair of indices by symmetry. This is only immediate for any pair of the first $k - 1$ indices. However, when we are considering an (i, k) -axis-aligned edge for some $i \in [k - 1]$, the probability of cutting this edge can only be smaller as there is one less threshold to cut it; see [17, Remark 2] for a discussion on this aspect. This reasoning holds even when there is no terminal at the k th vertex of the simplex.

Thus, the rounding scheme of [5], with the modifications of using Algorithm 1 rather than exponential clocks and only permuting the first q coordinates, gives an α -approximation for LIFTED CUT. For completeness, we recall the algorithms and lemmas from [5], with the appropriate modifications, in Appendix A. \square

5. Single-to-All Problem

In this problem, we are looking for a *single* representative from each candidate set that will be separated from *every* candidate in the other candidate sets. This includes the other representatives, making the problem very similar to MULTIWAY CUT once the representatives are chosen. However, a key difference is that the optimal partition may have $q + 1$ components.

We first look at the case where q is constant.

Theorem 8. *For fixed q , there is a polynomial-time α -approximation algorithm for SINGLE-TO-ALL.*

Proof. First, guess the representative t_i for each $i \in [q]$. As there are only $\prod_{i=1}^q |T_i| \leq n^q$ possible choices, this is polynomial in n for fixed q . For a fixed choice of representatives, SINGLE-TO-ALL is an instance of LIFTED CUT. To see this, fix the labels $\ell(v)$ for $v \in V$ as follows:

- (a) If $v = t_i$ for some i , set $\ell(v) := \{i\}$.
- (b) Otherwise, if $v \in T_i \setminus \{t_i\}$, set $\ell(v) := \{i, q+1\}$.
- (c) Finally, if $v \in V \setminus \bigcup_{i \in [q]} T_i$, set $\ell(v) := \{1, \dots, q+1\}$.

Condition 1 in the definition of LIFTED CUT is a clear consequence of rule (a), and since any node that is not a representative can be labeled $q+1$ by rules (b) and (c), Condition 2 follows as well. Therefore, Theorem 7 leads to an α -approximation. \square

Following the idea of the classical 2-approximation for MULTIWAY CUT discussed in the introduction, there is a simple 2-approximation when q is arbitrary.

Theorem 9. *There is a polynomial-time 2-approximation algorithm for SINGLE-TO-ALL.*

Proof. For each candidate set T_i , let $t_i \in T_i$ be a node for which the minimum-weight cut separating t_i from $\cup_{j \neq i} T_j$ is as small as possible, and let C be the union of these isolating cuts. To see that the solution is within a factor 2 of the optimum, consider an optimal solution to SINGLE-TO-ALL and let V_1, \dots, V_{q+1} denote the components after its deletion, where V_{q+1} may be empty and the components are ordered by the indices of the representatives they contain. The boundary of each V_i is an isolating cut of some candidate in T_i , which the algorithm minimized. Summing up the weights of the boundaries, we count each edge twice, and the theorem follows. \square

6. Algorithms when q is a Fixed Constant

In this section, we study the FIXED-TO-SINGLE, SOME-TO-SINGLE, SOME-TO-SOME, and SOME-TO-ALL problems in the regime where q is a fixed constant. Although the problems seem similar, the techniques used for a fixed q differ significantly, suggesting fundamental differences between them.

6.1. FIXED-TO-SINGLE

The FIXED-TO-SINGLE problem is slightly different from the others, as the goal here is to choose representatives that need to be separated only from a *fixed* node s . In this case, the problem can be solved efficiently.

Theorem 10. *For fixed q , FIXED-TO-SINGLE can be solved in polynomial time.*

Proof. In this case, one can iterate through all possible choices of representatives, of which we have at most n^q , calculate a minimum two-way $s - \{t_i : i \in [q]\}$ cut for each, and then take the best of all solutions. \square

6.2. SOME-TO-SINGLE and SOME-TO-SOME

The SOME-TO-SINGLE and SOME-TO-SOME problems both become MULTICUT instances with a constant number of terminals, which leads the following theorem.

Theorem 11. *For fixed q , there is a polynomial-time α -approximation algorithm to SOME-TO-SINGLE and SOME-TO-SOME.*

Proof. We will use the α -approximation to MULTIWAY CUT on a polynomial number of instances with fixed terminals.

We begin with the SOME-TO-SINGLE problem. In this problem, the goal is to choose a single representative t_j for each $j \in [q]$ together with some candidate $t_i^j \in T_i$ for each pair $i \neq j$ that are then separated by the cut. When q is fixed, one can guess the representatives t_i^j and t_j to get a set of terminals S together with some separation demands on them. The number of such terminals can be bounded by $|S| \leq q^2$. Therefore, the number of guesses for S is polynomial in n . Each guess of S defines a minimum multicut problem since we know which pairs of representatives have to be separated. We can compute an α -approximation to each of these MULTICUT problems by enumerating all possible partitions of S that satisfy the multicut demands, contracting the partitions into fixed terminals, and calculating an α -approximating multiway cut for each.

For the SOME-TO-SOME problem, once can guess the representatives t_i^j for each $i, j \in [q], i \neq j$ to get a set of terminals S together with some separation demands on them. Since any candidate set with terminals in different components already has at least one element in a separate component for any other candidate set, the number of such terminals can be bounded by $|S| \leq 2q$. For each fixed S , we can find an α -approximation the same way as above. \square

Remark 12. Combining the approximation for SOME-TO-SOME with Theorem 17 gives the current best approximation for STEINER MULTICUT in the regime where the number of candidates depends on n , and the number of sets is constant.

6.3. SOME-TO-ALL

Finally, we consider the SOME-TO-ALL problem, which asks to find representatives $t_i^j \in T_i$ for each pair $i, j \in [q]$ and a minimum-weight cut $C \subseteq E$ such that t_i^j is separated from *all* of T_j in $G - C$. The case for constant q uses the tool from Section 4.

Theorem 13. *For fixed q , there is a polynomial-time α -approximation algorithm for SOME-TO-ALL.*

Proof. After guessing the n^{q^2} possible representatives t_i^j , we also guess a valid partition V_1, \dots, V_{q_1} of these representatives into q_1 components, where $2 \leq q_1 \leq q^2$. The number of such partitions is exponential in q , but we can still enumerate them when q is fixed. For such a partition, the problem becomes an instance of $(q_1 + 1)$ -dimensional LIFTED CUT with the following labels.

- (a) If $v \in V_i$ for some i , set $\ell(v) := \{i\}$.
- (b) Otherwise, if $v \in T_j$, ensure the label cannot be any partition containing some t_i^j . In other words, set $\ell(v) := \{1, \dots, q_1 + 1\} \setminus \{k: v \in T_j \text{ and } t_i^j \in V_k \text{ for some } i, j\}$.
- (c) Finally, if $v \in V \setminus \bigcup_{i \in [q]} T_i$, $\ell(v) := \{1, \dots, q_1 + 1\}$.

Conditions 1 and 2 of LIFTED CUT are not difficult to verify. The solution to this problem is a solution to SOME-TO-ALL. Indeed, consider the partition given by a solution to LIFTED CUT, which is an extension of the partition V_1, \dots, V_{q_1} by rule (a), with an additional class for label $q_1 + 1$. Rule (b) then ensures, for a given t_i^j , that the component of t_i^j cannot contain any element of T_j . Thus, Theorem 7 gives an α -approximation for SOME-TO-ALL as well. \square

7. Hardness Results when q is Part of the Input

In this section, we provide hardness results for FIXED-TO-SINGLE, SOME-TO-SINGLE, SOME-TO-SOME, and SOME-TO-ALL when the number of terminal sets q is a part of the input.

In our proofs, we use reduction from HITTING SET, which is defined as follows: Given a family S_1, \dots, S_m of subsets of S and an integer k , the goal is to decide whether there exists a set $X \subseteq S$ with $|X| \leq k$ that intersects each S_i in at least one element.

Theorem 14. *FIXED-TO-SINGLE is at least as hard to approximate as HITTING SET.*

Proof. Let $\mathcal{S} = S_1, \dots, S_m$ be an instance of HITTING SET over a ground set V . Construct an instance of FIXED-TO-SINGLE by creating a weighted graph G with vertices $V \cup \{s\}$ and edge set $E = \{sv: v \in V\}$, each with weight 1. The graph G is then a star with center s , and let the candidate sets be exactly those in \mathcal{S} . Then, minimizing the number of edges needed to separate at least one node of each S_i from s is equivalent to finding a minimum hitting set. Given a hitting

set X , the cut $C = \{sx : x \in X\}$ gives a valid FIXED-TO-SINGLE solution of the same weight. Given a FIXED-TO-SINGLE solution with representatives t_1, \dots, t_q , the representatives form a solution to HITTING SET. Note that the reduction is approximation factor preserving, hence the theorem follows by Proposition 1. \square

Theorem 15. *SOME-TO-SINGLE is at least as hard to approximate as FIXED-TO-SINGLE.*

Proof. Let s, T_1, \dots, T_q be an instance of FIXED-TO-SINGLE on some edge weighted graph $G = (V, E)$. We create an instance $T'_1, \dots, T'_q, T'_{q+1}$ of SOME-TO-SINGLE on the same graph G with $T'_i := T_i \cup \{s\}$ for $i \in [q]$, and $T'_{q+1} := \{s\}$.

Given a solution to the FIXED-TO-SINGLE instance, we can obtain a solution of the same weight to the SOME-TO-SINGLE instance by keeping the representatives t_i for $i \in [q]$, setting $t_{q+1} = t'_{q+1} := s$ for $i \in [q]$, $t_i^{q+1} := t_i$ for $i \in [q]$, and $t_i^j := s$ for $i, j \in [q], i \neq j$.

For the other direction, we observe that t_j must be separated from s for $j \in [q]$ in a solution of the SOME-TO-SINGLE instance. Thus, we obtain a solution with the same weight for the FIXED-TO-SINGLE if we keep the same representatives. \square

Theorem 16. *SOME-TO-ALL is at least as hard to approximate as FIXED-TO-SINGLE.*

Proof. Let T_1, \dots, T_q be the terminal sets of an instance of FIXED-TO-SINGLE on a weighted graph $G = (V, E)$ where $q \geq 2$. We construct a SOME-TO-ALL instance by adding additional nodes $V_0 := \{s_1, \dots, s_q\}$, and letting $G' := (V \cup V_0, E)$ with the same edge weights. The new terminal sets are $T'_i = T_i \cup \{s_i\}$ for $i \in [q]$, and $T'_{q+1} = \{s, s_1, \dots, s_q\}$.

Take a FIXED-TO-SINGLE solution t_1^*, \dots, t_q^* . Then, we can construct a solution of the corresponding SOME-TO-ALL instance having the same weight by setting, for $i, j \in [q+1]$ with $i \neq j$,

$$t_i^j = \begin{cases} s_1 & \text{if } i = q+1, j \neq 1 \\ s_2 & \text{if } i = q+1, j = 1 \\ s_i & \text{if } i \neq q+1, j \neq q+1, \\ t_i^* & \text{if } i \neq q+1, j = q+1. \end{cases} .$$

Indeed, the same cut will separate each t_i^j from all of T'_j , and have the same weight.

Given an optimal solution to the SOME-TO-ALL instance, we can assume without loss of generality that the t_i^j are elements of $V_0 \setminus \{s_j\}$ when $i = q+1$, and $t_i^j = s_i$ when $i \neq q+1, j \neq q+1$, as these are separated from the corresponding T'_j in G' without incurring any extra cost. Then we can get a FIXED-TO-SINGLE

solution by setting $t_j = t_j^{q+1}$ for all $j \in [q]$ and removing the same edges. This reduction preserves approximation, as the solutions have the same weight. \square

The result for the SOME-TO-SOME problem is a little different, as it turns out to be equivalent to STEINER MULTICUT. Recall that in STEINER MULTICUT we are given a graph $G = (V, E)$, a weight function $w: E \rightarrow \mathbb{R}_+$ and subsets $X_1, \dots, X_q \subseteq V$, and the goal is to find a subset C of edges of minimum weight such that X_i intersects at least two components of $G - C$ for $i \in [q]$.

Theorem 17. *SOME-TO-SOME and STEINER MULTICUT are polynomial-time equivalent, meaning that any of them is polynomial-time reducible to the other.*

Proof. Let us consider an instance of STEINER MULTICUT, that is, we are given q subsets X_1, X_2, \dots, X_q of nodes of a graph G , each of which needs to be cut into at least two components. We construct a SOME-TO-SOME instance on the same graph with $2q$ candidate sets T_1, T_2, \dots, T_{2q} , where $T_i = X_{\lfloor(i+1)/2\rfloor}$ for $1 \leq i \leq 2q$. Then, for each $j = 1, \dots, q$, the condition that T_{2j-1} must be separated from T_{2j} ensures that there are two nodes $t_{2j-1}^{2j}, t_{2j}^{2j-1} \in X_j$ that are in different components. In other words, the solution is a minimal cut that, once removed, divides each set into at least two components. If the conditions of SOME-TO-SOME hold for T_{2j-1} and T_{2j} for any j , then they hold automatically for any other pair of candidate sets too, because once a set has elements in two components, at least one of them will be in a different component than some element of any given candidate set.

For the other direction, we are given q subsets T_1, \dots, T_q of nodes of a graph G as a SOME-TO-SOME instance. We then make a STEINER MULTICUT instance with $\binom{q}{2}$ vertex sets indexed by pairs $i, j \in [q]^2$ with $i \neq j$. The set $X_{i,j}$ will then be $T_i \cup T_j$, which means any valid STEINER MULTICUT solution C will split each of these $T_i \cup T_j$ sets into at least two components. We claim C is a valid SOME-TO-SOME solution as well. Let $v_{i,j}, u_{i,j} \in X_{i,j}$ be in different components of $G \setminus C$. Then one of the following cases must hold:

1. $v_{i,j} \in T_i$ and $u_{i,j} \in T_j$. In this case, let $t_j^i := u_{i,j}$ and $t_i^j := v_{i,j}$.
2. $u_{i,j} \in T_i$ and $v_{i,j} \in T_j$. In this case, let $t_j^i := v_{i,j}$ and $t_i^j := u_{i,j}$.
3. $u_{i,j}, v_{i,j} \in T_i$. Then either
 - (a) all of T_j is in the same component of $G \setminus C$ as $u_{i,j}$, in which case let $t_i^j := v_{i,j}$, and set t_j^i to an arbitrary element of T_j , or
 - (b) some vertex $w \in T_j$ is in a different component of $G \setminus C$ than $u_{i,j}$, in which case let $t_j^i := w$, and $t_i^j := u_{i,j}$.
4. Similarly, if $u_{i,j}, v_{i,j} \in T_j$, then either

- (a) all of T_i is in the same component of $G \setminus C$ as $u_{i,j}$, in which case let $t_j^i := v_{i,j}$, and set t_i^j to an arbitrary element of T_i , or
- (b) some vertex $w \in T_i$ is in a different component of $G \setminus C$ than $u_{i,j}$, in which case let $t_i^j := w$, and $t_j^i := u_{i,j}$.

In all cases above, t_j^i is in a different component than t_i^j on $G \setminus C$, so C is a valid SOME-TO-SOME solution. Any SOME-TO-SOME solution is clearly also a solution for this STEINER MULTICUT instance, so the optimal cut is the same for both. \square

References

- [1] K. Bérczi, T. Király, D. P. Szabo, Multiway cuts with a choice of representatives, in: 49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024, pp. 25–1.
- [2] K. Bérczi, T. Király, D. P. Szabo, Multiway cuts with a choice of representatives, arXiv preprint arXiv:2407.03877 (2024).
- [3] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, M. Yannakakis, The complexity of multiterminal cuts, SIAM Journal on Computing 23 (4) (1994) 864–894. [doi:10.1137/S0097539792225297](https://doi.org/10.1137/S0097539792225297).
- [4] G. Călinescu, H. Karloff, Y. Rabani, An improved approximation algorithm for multiway cut, Journal of Computer and System Sciences 60 (3) (2000) 564–574. [doi:https://doi.org/10.1006/jcss.1999.1687](https://doi.org/10.1006/jcss.1999.1687).
- [5] A. Sharma, J. Vondrák, Multiway cut, pairwise realizable distributions, and descending thresholds, in: Proceedings of the forty-sixth annual ACM symposium on Theory of computing, 2014, pp. 724–733.
- [6] K. Bérczi, K. Chandrasekaran, T. Király, V. Madan, Improving the integrality gap for multiway cut, Mathematical Programming 183 (1-2) (2020) 171–193. [doi:10.1007/s10107-020-01485-2](https://doi.org/10.1007/s10107-020-01485-2).
- [7] J. Kleinberg, E. Tardos, Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields, Journal of the ACM (JACM) 49 (5) (2002) 616–639.
- [8] C. Chekuri, S. Khanna, J. Naor, L. Zosin, Approximation algorithms for the metric labeling problem via a new linear programming formulation., in: SODA, Vol. 1, 2001, pp. 109–118.

- [9] H. Saran, V. V. Vazirani, Finding k cuts within twice the optimal, SIAM Journal on Computing 24 (1) (1995) 101–108. [doi:10.1137/S0097539792251730](https://doi.org/10.1137/S0097539792251730).
- [10] P. Manurangsi, Inapproximability of maximum biclique problems, minimum k-cut and densest at-least-k-subgraph from the small set expansion hypothesis, Algorithms 11 (1) (2018) 10.
- [11] C. Chekuri, S. Guha, J. Naor, The Steiner k-cut problem, SIAM Journal on Discrete Mathematics 20 (1) (2006) 261–271.
- [12] P. N. Klein, S. A. Plotkin, S. Rao, E. Tardos, Approximation algorithms for steiner and directed multicut, Journal of Algorithms 22 (2) (1997) 241–269.
- [13] A. Gupta, V. Nagarajan, R. Ravi, An improved approximation algorithm for requirement cut, Operations Research Letters 38 (4) (2010) 322–325. [doi: https://doi.org/10.1016/j.orl.2010.02.009](https://doi.org/10.1016/j.orl.2010.02.009).
- [14] R. Schwartz, Y. Shiloni, Approximating Requirement Cut via a Configuration LP, in: J. Byrka, R. Meka (Eds.), Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020), Vol. 176 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2020, pp. 53:1–53:16. [doi:10.4230/LIPIcs.APPROX.RANDOM.2020.53](https://doi.org/10.4230/LIPIcs.APPROX.RANDOM.2020.53).
- [15] D. Bertsimas, J. N. Tsitsiklis, Introduction to linear optimization, Vol. 6, Athena scientific Belmont, MA, 1997.
- [16] D. R. Karger, P. Klein, C. Stein, M. Thorup, N. E. Young, Rounding algorithms for a geometric embedding of minimum multiway cut, Mathematics of Operations Research 29 (3) (2004) 436–461. [doi:10.1287/moor.1030.0086](https://doi.org/10.1287/moor.1030.0086).
- [17] N. Buchbinder, J. Seffi, R. Schwartz, Simplex partitioning via exponential clocks and the multiway-cut problem, SIAM Journal on Computing 47 (4) (2018) 1463–1482.
- [18] I. Dinur, D. Steurer, Analytical approach to parallel repetition, in: Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC ’14, Association for Computing Machinery, New York, NY, USA, 2014, p. 624–633. [doi:10.1145/2591796.2591884](https://doi.org/10.1145/2591796.2591884).
- [19] D. Moshkovitz, The projection games conjecture and the np-hardness of ln n-approximating set-cover, Theory of Computing 11 (7) (2015) 221–235. [doi: 10.4086/toc.2015.v011a007](https://doi.org/10.4086/toc.2015.v011a007).

- [20] S. Kapoor, On minimum 3-cuts and approximating k-cuts using cut trees, in: International Conference on Integer Programming and Combinatorial Optimization, Springer, 1996, pp. 132–146.

Appendix A. Details of the Threshold Algorithms

For completeness, we include a detailed description of the α -approximation algorithm for lifted cut. This is just a collection of the results of Sharma and Vondrák [5], but understanding this is necessary for the proof of Theorem 7. The content of this section can be found in more detail in [5], the only modifications we make is to perform the rounding in $q + 1$ dimensions and make more clear the role of the $(q + 1)$ st vertex, as well as using Algorithm 1 instead of the exponential clocks algorithm.

First we describe the three threshold rounding schemes: the Single Threshold, Descending Thresholds, and Independent Thresholds schemes. These are described in Algorithms 4, 5, and 6, respectively. Each scheme is given a solution to LIFT-LP, and rounds it to an integer solution by assigning vertices to terminals. The Single Threshold Scheme takes as input some distribution with probability density ϕ , Descending Thresholds some distribution with density ψ , and Independent Thresholds with density ξ . Finally, these schemes are combined with appropriate parameters along with Algorithm 1 according to Algorithm 7, which takes additionally parameters $b, p_1, p_2, p_3, p_4 \in [0, 1]$, along with some probability density ϕ .

Algorithm 4 The Single Threshold Rounding Scheme

- 1: Choose threshold $\theta \in [0, 1)$ with probability density $\phi(\theta)$.
 - 2: Choose a random permutation σ of $[q]$.
 - 3: **for** $i = 1$ to q **do**
 - 4: For any unassigned $u \in V$ with $x_{\sigma(i)}^u \geq \theta$, assign u to terminal $\sigma(i)$.
 - 5: **end for**
 - 6: Assign all remaining unassigned vertices to terminal $q + 1$
-

The following three Lemmas are key to the analysis of Algorithm 7. The cut density for an edge of type (i, j) located at $(u_1, \dots, u_{q+1}) \in \Delta_{q+1}$ is the limit of the probability that the given threshold scheme assigns (u_1, \dots, u_{q+1}) and $(u_1, \dots, u_i + \varepsilon, \dots, u_j - \varepsilon, \dots, u_{q+1})$ to different terminals, normalized by ε as $\varepsilon \rightarrow 0$.

Lemma 18 (Lemma 5.1 in [5]). *Given a point $(u_1, \dots, u_{q+1}) \in \Delta_{q+1}$ and the parameter b of Algorithm 7, let $a = \frac{1-u_i-u_j}{b}$. If $a > 0$, the cut density for an edge*

Algorithm 5 Descending Thresholds Rounding Scheme

- 1: For each $i \in [q]$, independently choose threshold $\theta_i \in [0, 1)$ with probability density $\psi(\theta)$.
 - 2: Choose a random permutation σ of $[q]$ such that $\theta_{\sigma(1)} \geq \dots \geq \theta_{\sigma(q)}$.
 - 3: **for** $i = 1$ to q **do**
 - 4: For any unassigned $u \in V$ with $x_{\sigma(i)}^u \geq \theta_{\sigma(i)}$, assign u to terminal $\sigma(i)$.
 - 5: **end for**
 - 6: Assign all remaining unassigned vertices to terminal $q + 1$
-

Algorithm 6 Independent Threshold Rounding Scheme

- 1: For each $i \in [q]$, independently choose threshold $\theta_i \in [0, 1)$ with probability density $\xi(\theta)$.
 - 2: Choose a uniformly random permutation σ of $[q]$.
 - 3: **for** $i = 1$ to q **do**
 - 4: For any unassigned $u \in V$ with $x_{\sigma(i)}^u \geq \theta_{\sigma(i)}$, assign u to terminal $\sigma(i)$.
 - 5: **end for**
 - 6: Assign all remaining unassigned vertices to terminal $q + 1$
-

Algorithm 7 Sharma-Vondrák Rounding Scheme

- 1: With probability p_1 , choose the Kleinberg-Tardos Rounding Scheme (Algorithm 1).
 - 2: With probability p_2 , choose the Single Threshold Rounding Scheme (Algorithm 4) with probability density ϕ .
 - 3: With probability p_3 , choose the Descending Threshold Rounding Scheme (Algorithm 5), where the thresholds are chosen uniformly in $[0, b]$.
 - 4: With probability p_4 , choose the Independent Threshold Rounding Scheme (Algorithm 6), where the thresholds are chosen uniformly in $[0, b]$.
-

of type (i, j) , where $i \neq j$ are indices in $[q + 1]$ located at (u_1, \dots, u_{q+1}) under the Independent Thresholds Rounding Scheme with parameter b is at most

- $\frac{2(1-e^{-a})}{ab} - \frac{(u_i+u_j)(1-(1+a)e^{-1})}{a^2b^2}$, if all the coordinates u_1, \dots, u_{q+1} are in $[0, b]$.
- $\frac{(a+e^{-a}-1)}{a^2b}$, if $u_i \in [0, b], u_j \in (b, 1]$ and $u_\ell \in [0, b]$ for all other $\ell \in [q] \setminus \{i, j\}$.
- $\frac{1}{b} - \frac{(u_i+u_j)}{6b^2}$, if $u_i, u_j \in [0, b]$ and $u_\ell \in (b, 1]$ for some other $\ell \in [q] \setminus \{i, j\}$.

- $\frac{1}{3b}$, if $u_i \in [0, b], u_j \in (b, 1]$ and $u_\ell \in [0, b]$ for some other $\ell \in [q] \setminus \{i, j\}$.
- 0, if $u_i, u_j \in (b, 1]$.

For $a = 0$, the cut density is given by the limit of the expressions above as $a \rightarrow 0$.

Lemma 19 (Lemma 5.2 in [5]). *For an edge of type (i, j) located at (u_1, \dots, u_{q+1}) , where $i \neq j$ are indices in $[q + 1]$, the cut density under the Single Threshold Rounding Scheme is at most*

- $\frac{1}{2}\phi(u_i) + \phi(u_j)$, if $u_\ell \leq u_i \leq u_j$ for all other $\ell \in [q] \setminus \{i, j\}$.
- $\frac{1}{3}\phi(u_i) + \phi(u_j)$, if $u_i < u_\ell \leq u_j$ for some other $\ell \in [q] \setminus \{i, j\}$.
- $\frac{1}{2}\phi(u_i) + \phi(u_j)$, if $u_i \leq u_j < u_\ell$ for some other $\ell \in [q] \setminus \{i, j\}$.

Lemma 20 (Lemma 5.3 in [5]). *For an edge of type (i, j) located at (u_1, \dots, u_{q+1}) , where $i \neq j$ are indices in $[q + 1]$, the cut density under the Descending Thresholds Rounding Scheme is at most*

- $(1 - \int_{u_i}^{u_j} \psi(u) du)\psi(u_i) + \psi(u_j)$, if $u_\ell \leq u_i \leq u_j$ for all other $\ell \in [q] \setminus \{i, j\}$.
- $(1 - \int_{u_i}^{u_j} \psi(u) du)((1 - \int_{u_i}^{u_\ell} \psi(u) du))\psi(u_i) + \psi(u_j)$, if $u_i < u_\ell \leq u_j$ for some other $\ell \in [q] \setminus \{i, j\}$.
- $(1 - \int_{u_i}^{u_j} \psi(u) du)(1 - \int_{u_i}^{u_\ell} \psi(u) du)\psi(u_i) + (1 - \int_{u_j}^{u_\ell} \psi(u) du)\psi(u_j)$, if $u_i \leq u_j < u_\ell$ for some other $\ell \in [q] \setminus \{i, j\}$.

The proof for each of these lemmas is exactly as in [5], except for one additional trivial observation: the cut density of an edge of type $(i, q + 1)$ is at most that of an edge of type (i, j) for any $j \neq i, j \neq q + 1$. This is because the $(q + 1)$ st terminal is considered last, and has no threshold of its own, and therefore cannot increase the separation probability. With these Lemmas in hand, Theorem 5.6 of [5] shows, with a specific choice of parameters, that Algorithm 7 is a 1.2965-approximation to LIFTED CUT as well.