Multiway Cuts with a Choice of Representatives

Kristóf Bérczi
MTA-ELTE Matroid Optimization Research Group, Budapest, Hungary
HUN-REN–ELTE Egerváry Research Group on Combinatorial Optimization,
Department of Operations Research, Eötvös Loránd University, Budapest, Hungary

Tamás Király
HUN-REN–ELTE Egerváry Research Group on Combinatorial Optimization,
Department of Operations Research, Eötvös Loránd University, Budapest, Hungary

Daniel P. Szabo
Department of Operations Research, Eötvös Loránd University, Budapest, Hungary

Abstract

In the Multiway Cut problem, we are given an undirected graph with nonnegative edge weights and a subset of k terminals, and the goal is to determine 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 an extensive line of research has concentrated on closing the gap between the best upper and lower bounds for approximability and inapproximability, respectively.

In this paper, we study several generalizations of Multiway Cut where the terminals can be chosen as *representatives* from sets of *candidates* $T_1, \ldots, T_q$. In this setting, one is allowed to choose these representatives so that the minimum-weight cut separating these sets *via their representatives* is as small as possible. We distinguish different cases depending on (A) whether the representative of a candidate set has to be separated from the other candidate sets completely or only from the representatives, and (B) whether there is a single representative for each candidate set or the choice of representative is independent for each pair of candidate sets.

For fixed q, we give approximation algorithms for each of these problems that match the best known approximation guarantee for Multiway Cut. Our technical contribution is a new extension of the CKR relaxation that preserves approximation 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 give a 2-approximation algorithm for the case where we need to choose a single representative from each candidate set. This is a generalization of the $(2 - 2/k)$-approximation for k-Cut, and we can solve it by relating the tree case to optimization over a gammoid.

2012 ACM Subject Classification Theory of computation → Rounding techniques; Theory of computation → Facility location and clustering; Theory of computation → Network optimization; Theory of computation → Linear programming; Theory of computation → Graph algorithms analysis

Keywords and phrases Approximation algorithms, Multiway cut, CKR relaxation, Steiner multicut

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.25

Funding The research was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, by the Ministry of Innovation and Technology of Hungary – grant number ELTE TKP 2021-NKTA-62, and by Dynasnet European Research Council Synergy project – grant number ERC-2018-SYG 810115.

1 Introduction

For an undirected graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}_+$, the Multiway Cut problem asks for a minimum-weight cut $C \subseteq E$ separating any pair of terminals in a given terminal set $S = \{s_1, \ldots s_k\}$. As cuts can be identified with partitions of the nodes, 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.

1.1 Previous work

Dahlhaus et al. [5] 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 all other 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 [18] with an approximation factor of 1.2965, while the best known lower bound (assuming the Unique Games Conjecture) is slightly above 1.2 [2].

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 $(s_1, t_1), \ldots, (s_k, t_k)$, and the goal is to determine a minimum-weight cut whose removal disconnects each s_i from its pair t_i. Multicut is NP-hard to approximate within any constant factor assuming the Unique Games Conjecture [3], and there is a polynomial-time $O(\log k)$-approximation algorithm [7]. 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 edges; 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 [12] 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 [16] that is tight [13]. The Steiner Multicut [11] problem takes as input an undirected graph G and subsets $X_1, X_2, \ldots, X_q$ 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 [9], 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 [9, 17], of which we will use the $O(\log k \log q)$ approximation, where $k = |\bigcup_{i=1}^{q} X_i| \leq n$.

1.2 Our results

We introduce generalizations of Multiway Cut, where we are allowed to choose *representatives* from some terminal candidate sets $T_1, \ldots, T_q \subseteq V$, and the goal is to find the minimum-weight cut separating these sets *via their representatives*. The 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, we use the following naming rules.

- 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, Single-to-All 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 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. In the Fixed-to-Single problem, we are given a fixed node s, and we want 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. Although, as we will discuss later, some of our problems are equivalent or closely related to problems that have already been considered in the literature, a systematic study of this type of generalization has not yet been done, and some of our results (Theorem 3, Theorem 9) 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 [18]. 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.

1.3 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. 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.

Table 1 A summary of our results, where $\alpha \approx 1.2965$ [18] is the current best approximation factor for Multiway Cut. The tightness of 2-approximation assumes SSEH, while the other inapproximability results hold assuming P $\neq$ NP. The main results are highlighted in bold.

Problem	Demands	Fixed q	Unbounded q
All-to-All	$T_i - T_j$	α-approx	α-approx
Single-to-All	$t_i - T_j$	**α-approx**	2-approx
Single-to-Single	$t_i - t_j$	α-approx	**Tight 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 [9]
Some-to-All	$t_i^j - T_j$	**α-approx**	No $o(\log q)$ approx

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 3) in such a way that the rounding methods used in [18] 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. 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.

1.4 Structure of the paper

In Section 2, we present the main tools used in our algorithms and proofs. Section 3 introduces the Lifted Cut problem and describes how to extend the α-approximation of [18] to Lifted Cut. The remaining sections present the results for the problems listed in Table 1.

2 Background

Throughout the paper, we denote the set of non-negative reals by $\mathbb{R}_+$, and use $[k] = \{1, \dots, k\}$. We use e^i to denote ith elementary vector, and Δ_k denotes the convex hull of $\{e^1, \dots, e^k\}$, that is, $\Delta_k = \{x \in \mathbb{R}^k : x \geq 0, \ \sum_{i=1}^k x_i = 1\}$.

Given an undirected graph $G = (V, E)$, the edge going between nodes $u, v \in V$ is denoted by (u, v). For a weight function $w : E \to \mathbb{R}_+$ 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 $\delta(S) = \{(u, v) \in E : u \in S, v \in V \setminus S\}$.

We briefly summarize the background results that we build upon in our proofs.

2.1 The CKR Relaxation and Rounding Methods

For a graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_+$ and terminals $S = \{s_1, \ldots, s_k\}$, the CKR relaxation [4] is the following linear program (CKR-LP) which assigns to each node $u \in V$ a geometric location x^u in the k-dimensional simplex.

$$\begin{aligned} \text{minimize} \quad & \sum_{(u,v)\in E} w_{u,v}\|x^u - x^v\|_1 \\ \text{subject to} \quad & x^u \in \Delta_k \quad u \in V, \\ & x^{s_i} = e^i \quad\ \ i \in [k]. \end{aligned} \tag{CKR-LP}$$

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 take one of the two permutations $\sigma = (1, \ldots, k-1, k)$ and $(k-1, k-2, \ldots, 1, k)$ of the terminals at random (that is, with probability 1/2), 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 a permutation of the terminals and then assigns the nodes within some threshold to the terminals in that order a *threshold algorithm*. The analyses 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 u, v 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 [10, 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 [1]. 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 [12] 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 (UML-LP) 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.

$$\begin{aligned} \text{minimize} \quad & \sum_{(u,v)\in E} w_{u,v}\|x^u - x^v\|_1 \\ \text{subject to} \quad & x^u \in \Delta_q \quad u \in V, \\ & x^v_i = 0 \quad\ \ i \notin \ell(v). \end{aligned} \tag{UML-LP}$$

It is shown in [1, Section 6] that Algorithm 1 gives the same guarantees as the exponential clocks algorithm.

The approximation algorithm of Sharma and Vondrák [18] for Multiway Cut randomly chooses between four different algorithms of the above two types with some careful analysis to achieve an α-approximation, where $\alpha \approx 1.2965$.

Algorithm 1 The Kleinberg-Tardos Algorithm for Uniform Metric Labeling.

Input: A graph $G = (V, E)$, weights $w : E \to \mathbb{R}_+$, labels $\ell : V \to \mathcal{P}([q])$, and an LP solution x^u for each $u \in V$.
Output: A solution to Uniform Metric Labeling.

```
while ∃u ∈ V s.t. u is unassigned do
    Pick a label i ∈ [q] uniformly at random, and a threshold ρ ~ unif[0, 1].
    Assign label i to any unassigned u ∈ V with x_i^u ≥ ρ.
end while
```

2.2 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 proved by Dinur and Steurer [6]. 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 [13]. For completeness, we include the relevant theorems here.

▶ **Theorem 1** ([6, 14])**.** *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*.*

▶ **Theorem 2** ([13])**.** *Assuming the Small Set Expansion Hypothesis, it is NP-hard to approximate k-Cut to within $(2-\varepsilon)$ factor of the optimum, 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 $T \subseteq V \setminus S$. A set $X \subseteq T$ is independent in M if there exist $|X|$ node-disjoint paths from elements of S into X. Optimizing over a gammoid, as with any other matroid, can be done efficiently using the greedy algorithm.

Finally, Gomory-Hu (GH) tree [8] is a standard tool in graph cut algorithms. The *GH tree* of a graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}_+$ is a tree $T = (V, F)$ together with weight function $w_T : F \to \mathbb{R}_+$ 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 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 $w : E \to \mathbb{R}_+$, *fixed* terminals $S = \{s_1, s_2, \ldots, s_q\} \subseteq V$, and a list of possible colors for each node $\ell : V \to \mathcal{P}[q+1]$, the power set of $[q+1]$, satisfying the following two conditions:

(A) $\ell(s_i) = \{i\}$ for $i = 1 \ldots q$,
(B) $q + 1 \in \ell(v)$ 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 call the following linear programming relaxation of the Lifted Cut problem LIFT-LP:

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

Condition A ensures that the set S indeed defines terminals that vertices of the simplex are assigned to, as in Multiway Cut, but Condition B offers a relaxation, allowing a vertex of the simplex to not be assigned to any terminal. This condition gives an additional dimension to the simplex (see Figure 1), while still preserving the approximation guarantees given by the rounding algorithms for CKR.

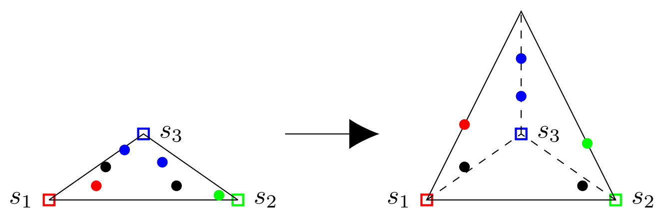

Figure 1 An example of the original CKR relaxation in relation to our extended LIFT-LP on the case for $t_i - T_j$. The point colors represent the different candidate sets.

▶ **Theorem 3.** *The rounding scheme of [18], 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.*

Proof. 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 ith component, since x^{s_i} is the ith vertex of the simplex. For other nodes $v \in V \setminus S$, $x^v_i = 0$ guarantees that v is not assigned to the ith 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 $x^v_{q+1} = 0$ 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 have 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 [18] carry through to this modified LIFT-LP. We give a sketch here, but the details are written out more carefully in Appendix B. We consider the two types of algorithms (i.e. threshold and exponential clocks) separately.

It was observed in [1] that the exponential clocks algorithm can be replaced by the 2-approximation for the Uniform Metric Labeling problem of Kleinberg-Tardos [12]. Since LIFT-LP corresponds to a Uniform Metric Labeling problem, the bound in [1, Lemma 3] remains valid in our case. Since this is the relevant 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 consider 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 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 [18] and [1] of the CKR relaxation for $k = q + 1$, as sketched below.

The threshold algorithms first choose a random 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 [18], thry prove results for the first two indices, and then argue that these hold for any pair of indices by symmetry. This is only directly clear for pairs in the first $k - 1$. However, when we consider 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 [1, Remark 2] for a discussion. This reasoning holds even when there is no terminal at the kth vertex of the simplex.

Thus, the rounding scheme of [18], 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 include the relevant algorithms and lemmas from [18], with the appropriate modifications, in Appendix B. ◀

4 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 other candidate sets. This includes the other representatives, making the problem very similar to Multiway Cut once the representatives are chosen. A key difference is that the optimal partition may have $q + 1$ components.

We first look at the case where q is constant.

▶ **Theorem 4.** *There is an α-approximation algorithm for Single-to-All when q is fixed.*

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. If a representative t_i is in T_j for some $j \neq i$, then there is obviously no solution. Otherwise, for a fixed choice of representatives, Single-to-All is an instance of Lifted Cut. To see this, observe that the problem is equivalent to the Uniform Metric Labeling problem obtained by fixing the labels $\ell(v)$ for $v \in V$ as follows:

1. If $v = t_i$ for some i, then set $\ell(v) := \{i\}$.
2. Otherwise, if $v \in T_i \setminus \{t_i\}$ for a unique i, then set $\ell(v) := \{i, q + 1\}$.
3. If $v \in T_i \cap T_j \setminus \{t_i, t_j\}$ for $i \neq j$, then set $\ell(v) := \{q + 1\}$.
4. Finally, if $v \in V \setminus \bigcup_{i\in[q]} T_i$, then set $\ell(v) := [q + 1]$.

This is an instance of Lifted Cut: Condition A is a clear consequence of the first rule, and since any node that is not a representative has $q + 1$ as one of its labels, Condition B follows as well. Therefore, Theorem 3 leads to an α-approximation. ◀

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 5.** *There is a 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

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

Input: A tree $G = (V, E)$, weights $w : E \to \mathbb{R}_+$, candidates $T_1, \ldots, T_q \subseteq V$.
Output: A minimum-weight good cut $C \subseteq E$.

```
1: Set C ← ∅.
2: while |C| < q − 1 do
3:     e ← arg min{w(e) : e ∉ C, C + e is good}
4:     C = C + e
5: end while
```

SINGLE-TO-ALL and let $V_1, V_2, \ldots, V_q, V_{q+1}$ denote the components after its deletion, where V_{q+1} may be empty and the components are ordered by the index of the representative 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. ◀

5 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 $T_1 = T_2 = \ldots = T_q = V$, SINGLE-TO-SINGLE generalizes k-CUT where we seek the minimum-weight cut that partitions the graph into k parts. It is known that k-CUT is hard to approximate within a factor of $2 - \varepsilon$ for any $\varepsilon > 0$, assuming SSEH [13].

▶ **Theorem 6.** *There is an α-approximation for SINGLE-TO-SINGLE when q is fixed.*

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. ◀

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 7.** *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, $S := \{s_1, \ldots, s_q\}$, and $T := V$, and consider the gammoid $M = (D, S, T)$. The key observation is the following.

▷ Claim 8. 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 $Z = \{v_1, \ldots, v_p\}$. Without loss of generality, we may assume that the candidate sets having a valid set of representatives in $G - C$ are $T_1, \ldots, T_p, T_{p+1}$, 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}.

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

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

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 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 $G - C$. ◁

By Claim 8, a minimum-weight good cut can be determined using the greedy algorithm for matroids, which is exactly what Algorithm 2 is doing. ◀

Algorithm 2 solves the special case when G is a tree. The classical $(2-2/k)$ approximation for MULTIWAY CUT [5] uses 2-way cuts coming from the Gomory-Hu tree, and so does the $(2-2/k)$ approximation for k-CUT [16]. 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. The algorithm is presented as Algorithm 3.

▶ **Theorem 9.** *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 $\delta(V_q^*)$. Let also H be the GH tree of G.

We transform OPT into a solution OPT_{GH} on H, losing at most a factor of $(2-2/q)$. We do this by repeatedly removing the minimum weight edge in $E(H)$ that separates a pair among the representatives $t_1^*, \dots, t_q^*$ that are in the same component of H. More precisely, we start with $H_0 = H$, and take the minimum-weight edge $e_1 \in E(H_0)$ separating some pair of representatives t_i^*, t_j^* in OPT that are in the same component of H_0. Define the edge $f_1 = (t_i^*, t_j^*)$. Then we construct $H_1 = H_0 - e_1$, and repeat this process to get a sequence of edges $e_1, e_2, \dots, e_{q-1}$ and a tree of representative pairs $F = (\{t_1^*, \dots, t_q^*\}, \{f_1, \dots, f_{q-1}\})$.

Direct the edges of F away from t_q^*, and reorder the edges such that f^1 is the edge going into t_1^*, f^2 into t_2^*, and so on. 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, and let $U(e^i)$ be the cut corresponding to e^i for each 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 these.

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

$$w(ALG) \leq w_H(ALG_{GH}) \leq w_H(OPT_{GH}) = \sum_{i=1}^{q-1} w(U(e^i)) \leq \sum_{i=1}^{q-1} w(\delta(V_i^*))$$

$$\leq (1-1/q) \sum_{i=1}^{q} w(\delta(V_i^*)) \leq (2-2/q) w(OPT).$$ ◀

6 Fixed-to-Single, Some-to-Single, Some-to-Some, and Some-to-All Problems

In this section, we combine the study of four problems, as the techniques are similar.

6.1 Hardness of approximation

All four have similar proofs of hardness of approximation, which we state here but leave the proofs to appendix A for brevity.

▶ **Theorem 10.** *For general q, Fixed-to-Single, Some-to-Single and Some-to-All are at least as hard to approximate as Hitting Set.*

We omit the Some-to-Some problem from Theorem 10 because it follows as a corollary to Theorem 11, which states that it is equivalent to the known Steiner Multicut problem. The conditional $o(\log n)$ inapproximability was already proved for Steiner Multicut in [15], using similar instances as those in our proof of Theorem 10. The Some-to-Some problem asks to find a cut such that each pair of candidate sets have at least one element in separate components, where this choice can depend on the pair.

▶ **Theorem 11.** *The Some-to-Some problem is equivalent to Steiner Multicut.*

Proof. To reduce from Steiner Multicut, we are given q subsets $X_0, X_1, \dots, X_{q-1}$ 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 $T_0, T_1, \dots, T_{2q-1}$, where $T_i = X_{\lfloor i/2 \rfloor}$ for $0 \le i \le 2q-1$. Then, for each $j = 0 \dots q-1$, the condition that T_{2j} must be separated from T_{2j+1} ensures that there are two nodes $t_{2j}^{2j+1}, t_{2j+1}^{2j} \in X_j$ 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} and T_{2j+1} 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 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
   - **(i)** 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
   - **(ii)** 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
   - **(i)** 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
   - **(ii)** 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. ◀

6.2 Fixed q

The techniques when q is fixed differ, suggesting that the problems themselves are quite different, despite the apparent similarities.

Fixed Terminal

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.

▶ **Proposition 12.** *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. ◀

Some to single/some

The Some-to-Single and Some-to-Some problems both become Multicut instances with a constant number of terminals in this case, which gives the following theorem:

▶ **Theorem 13.** *For fixed q, there is an α-approximation 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 as $|S| \leq q^2$. A slightly more careful analysis shows that the number of different t_i^j nodes for a candidate set T_i can be bounded by two. Thus, we only have to guess three representatives from each T_i, implying $|S| \leq 3q$. Either way, 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 (of which there are exponentially many in q) that satisfy the multicut demands, collapsing the partitions into fixed terminals, and calculating an α-approximating multiway cut for each.

For the Some-to-Some problem, again 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. ◀

Combining this approximation for Some-to-Some with Theorem 11 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.

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 3.

▶ **Theorem 14.** *There is an α-approximation for SOME-TO-ALL when q is fixed.*

Proof. We guess all representatives t_i^j; there are at most n^{q^2} possible choices, which is polynomial if q is fixed. Note that we may assume that $t_i^j \neq t_j^\ell$ if $i \neq j$, otherwise there is obviously no solution. We also guess a valid partition $V_1, \ldots, V_{q_1}$ of these representatives into q_1 components, where $2 \leq q_1 \leq q^2$ (validity means that t_i^j and t_j^ℓ are in different classes of the partition if $i \neq j$). The number of such partitions is exponential in q, but we can still enumerate them when q is fixed (note that this is not a partition of V, but a partition of the set of all chosen representatives, which is a vertex set of size at most q^2). For such a partition, the problem becomes an instance of $(q_1 + 1)$-dimensional LIFTED CUT with the following labels.

a) If $v \in V_k$ for some $k \in [q_1]$, then set $\ell(v) := \{k\}$.

b) Otherwise, if $v \in T_j$, then we must ensure that the label cannot be any partition containing some t_i^j. In other words, set $\ell(v) := \{1, 2, \ldots, q_1 + 1\} \setminus \{k : v \in T_j \text{ and } t_i^j \in V_k \text{ for some } i, j\}$.

c) Finally, if $v \in V \setminus \bigcup_{i \in [q]} T_i$, then set $\ell(v) := [q_1 + 1]$.

Conditions A and B 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 $V_1, \ldots, V_{q_1}$ by condition a, with an additional class for label $q_1 + 1$. Condition 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 3 gives an α-approximation for SOME-TO-ALL as well. ◀

References

1 Niv Buchbinder, Joseph (Seffi) Naor, and Roy Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 535–544, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2488608.2488675.

2 Kristóf Bérczi, Karthekeyan Chandrasekaran, Tamás Király, and Vivek Madan. Improving the integrality gap for multiway cut. *Mathematical Programming*, 183(1-2):171–193, 2020. doi:10.1007/s10107-020-01485-2.

3 Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D Sivakumar. On the hardness of approximating multicut and sparsest-cut. *computational complexity*, 15:94–114, 2006.

4 Gruia Călinescu, Howard Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. *Journal of Computer and System Sciences*, 60(3):564–574, 2000. doi:10.1006/jcss.1999.1687.

5 E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994. doi:10.1137/S0097539792225297.

6 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 624–633, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2591796.2591884.

7 Naveen Garg, Vijay V Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.

8 Ralph E Gomory and Tien Chung Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

9 Anupam Gupta, Viswanath Nagarajan, and R. Ravi. An improved approximation algorithm for requirement cut. *Operations Research Letters*, 38(4):322–325, 2010. doi:10.1016/j.orl.2010.02.009.

10 David R. Karger, Philip Klein, Cliff Stein, Mikkel Thorup, and Neal E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Mathematics of Operations Research*, 29(3):436–461, 2004. doi:10.1287/moor.1030.0086.

11 Philip N Klein, Serge A Plotkin, Satish Rao, and Eva Tardos. Approximation algorithms for steiner and directed multicuts. *Journal of Algorithms*, 22(2):241–269, 1997.

12 J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 14–23, 1999. doi:10.1109/SFFCS.1999.814572.

13 Pasin Manurangsi. Inapproximability of maximum biclique problems, minimum k-cut and densest at-least-k-subgraph from the small set expansion hypothesis. *Algorithms*, 11(1):10, 2018.

14 Dana Moshkovitz. The projection games conjecture and the np-hardness of ln n-approximating set-cover. In *APPROX-RANDOM*, pages 276–287. Springer, 2012.

15 Viswanath Nagarajan and R Ravi. Approximation algorithms for requirement cut on graphs. *Algorithmica*, 56:198–213, 2010.

16 Huzur Saran and Vijay V. Vazirani. Finding k cuts within twice the optimal. *SIAM Journal on Computing*, 24(1):101–108, 1995. doi:10.1137/S0097539792251730.

17 Roy Schwartz and Yotam Sharoni. Approximating Requirement Cut via a Configuration LP. In Jarosław Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 53:1–53:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.53.

18 Ankit Sharma and Jan Vondrák. Multiway cut, pairwise realizable distributions, and descending thresholds. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 724–733, 2014.

A Hardness Proofs for Fixed-to-Single, Some-to-Single, and Some-to-All

Here we prove Theorem 10. In the Hitting Set problem, the input is a family $S_1, \ldots, S_m$ of sets, and the goal is to find a smallest set of elements that intersects each of them.

▶ **Proposition 15.** *For general q, Fixed-to-Single is at least as hard to approximate as Hitting Set.*

Proof. To reduce Hitting Set to Fixed-to-Single, we create a graph G by adding a node s together with edges of weight 1 from s to the elements of the ground set. Thus we get a star with center s, and choose the candidate sets to be the sets $S_1, \ldots, S_m$. 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; see Figure 2 for an illustration. Note that the reduction is approximation factor preserving. Since there is no $o(\log m)$ approximation for Hitting Set assuming P $\neq$ NP [6, 14], the hardness of Fixed-to-Single follows. ◀

▶ **Proposition 16.** *For general q, Some-to-Single is at least as hard to approximate as Fixed-to-Single.*

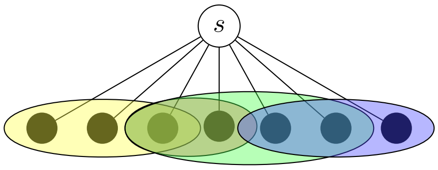

Figure 2 A picture of the reduction for FIXED-TO-SINGLE from HITTING SET.

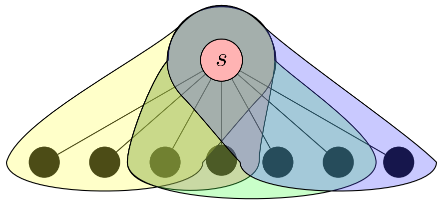

Figure 3 A picture of the reduction from FIXED-TO-SINGLE to SOME-TO-SINGLE.

Proof. Given an instance $s, T_1, \dots, T_q$ of FIXED-TO-SINGLE, we create an instance of SOME-TO-SINGLE with $T'_i \coloneqq T_i \cup \{s\}$ for $i \in [q]$, and $T'_{q+1} \coloneqq \{s\}$; see Figure 3 for an example.

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 $t_{q+1} = t^i_{q+1} \coloneqq s$ for $i \in [q]$, $t^{q+1}_i \coloneqq t_i$ for $i \in [q]$, and $t^j_i \coloneqq s$ for $i, j \in [q], i \neq j$.

For the other direction, we observe that each t_j $(j \in [q])$ must be separated from s 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 t_j $(j \in [q])$. ◀

▶ **Proposition 17.** *For general q, SOME-TO-ALL is at least as hard to approximate as FIXED-TO-SINGLE.*

Proof. Given an instance of FIXED-TO-SINGLE with sets $T_1, \dots, T_q$ on a graph $G = (V, E)$ where $q \geq 2$, we construct a SOME-TO-ALL instance as follows. We add additional nodes $V_0 = \{s_1, s_2, \dots, s_q\}$, $G' = (V \cup V_0, E)$, $T'_i = T_i \cup \{s_i\}$ for $i = 1 \dots q$, and $T'_{q+1} = \{s, s_1, s_2, \dots, s_q\}$; see Figure 4 for an example.

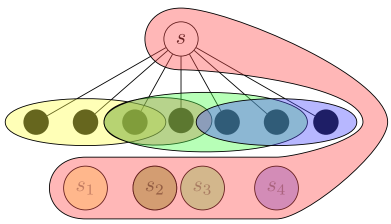

Figure 4 A picture of the reduction from FIXED-TO-SINGLE. The candidate set T_1 is in yellow, T_2 in brown, T_3 in green, T_4 in blue, and T_5 in red.

Given a FIXED-TO-SINGLE solution with representatives $t^*_1, \dots, t^*_q$, we get a solution to this instance as follows: $t^j_{q+1} = s_{(j+1) \bmod q}$ for $j \in [q+1]$; if $i \in [q]$, then $t^{q+1}_i = t^*_i$, and $t^j_i = s_i$ for $j \in [q]$. Then the same cut will separate each t^j_i 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 $t^j_i = s_{(j+1) \bmod q}$ when $i = q+1$, and $t^j_i = s_i$ when $i \neq q+1, j \neq q+1$, as these are separated from the corresponding T'_j in G'. Then we can get a FIXED-TO-SINGLE solution by setting $t_j = t^{q+1}_j$ for all $j \in [q]$ and removing the same edges. This reduction preserves approximation, as the solutions have the same weight. ◀

B 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 [18], but understanding this is necessary for the proof of Theorem 3. The content of this section can be found in more detail in [18], the only modifications we make are to perform the rounding in $k+1$ dimensions and make more clear the role of the $(k+1)$st vertex.

First we describe the three threshold rounding schemes: Single Threshold, Descending Thresholds, and Independent Thresholds. 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.

```
Choose threshold θ ∈ [0, 1) with probability density φ(θ).
Choose a random permutation σ of [k].
for all i ∈ [k] do
    For any unassigned u ∈ V with x^u_σ(i) ≥ θ, assign u to terminal σ(i).
end for
Assign all remaining unassigned vertices to terminal k + 1
```

Algorithm 5 Descending Thresholds Rounding Scheme.

```
For each i ∈ [k], choose threshold θ_i ∈ [0, 1) with probability density ψ(θ).
Choose a random permutation σ of [k] such that θ_σ(1) ≥ θ_σ(2) ≥ ... ≥ θ_σ(k).
for all i ∈ [k] do
    For any unassigned u ∈ V with x^u_σ(i) ≥ θ_σi, assign u to terminal σ(i).
end for
Assign all remaining unassigned vertices to terminal k + 1
```

Algorithm 6 Independent Threshold Rounding Scheme.

```
For each i ∈ [k], choose independently threshold θ_i ∈ [0, 1) with probability density ξ(θ).
Choose a uniformly random permutation σ of [k].
for all i ∈ [k] do
    For any unassigned u ∈ V with x^u_σ(i) ≥ θ_σ(i), assign u to terminal σ(i).
end for
Assign all remaining unassigned vertices to terminal k + 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 $(u_1, u_2, \dots, u_{k+1}) \in \Delta_{k+1}$ is the limit of the probability that the given threshold scheme assigns $(u_1, u_2, \dots, u_{k+1})$ and $(u_1 + \varepsilon, u_2 - \varepsilon, \dots, u_{k+1})$ to different terminals, normalized by ε as $\epsilon \to 0$.

Algorithm 7 The 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 $[0, b]$.

▶ **Lemma 18** (Lemma 5.1 in [18]). *Given a point $(u_1, u_2, \ldots, u_{k+1}) \in \Delta_{k+1}$ 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 of type (i, j), where $i \neq j$ are indices in $[k + 1]$ located at $(u_1, u_2, \ldots, u_{k+1})$ 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 $u_1, u_2, \ldots, u_{k+1}$ 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 $\ell \in [k] \setminus \{i, j\}$.*
- $\frac{1}{b} - \frac{(u_i+u_j)}{6b^2}$, *if $u_i, u_j \in [0, b]$ and $u_\ell \in (b, 1]$ for some other $\ell \in [k] \setminus \{i, j\}$.*
- $\frac{1}{3b}$, *if $u_i \in [0, b], u_j \in (b, 1]$ and $u_\ell \in [0, b]$ for some other $\ell \in [k] \setminus \{i, j\}$.*
- 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 $a \to 0$.

▶ **Lemma 19** (Lemma 5.2 in [18]). *For an edge of type (i, j) located at $(u_1, u_2, \ldots, u_{k+1})$, where $i \neq j$ are indices in $[k + 1]$, 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 $\ell \in [k] \setminus \{i, j\}$.*
- $\frac{1}{3}\phi(u_i) + \phi(u_j)$, *if $u_i < u_\ell \leq u_j$ for some other $\ell \in [k] \setminus \{i, j\}$.*
- $\frac{1}{2}\phi(u_i) + \phi(u_j)$, *if $u_i \leq u_j < u_\ell$ for some other $\ell \in [k] \setminus \{i, j\}$.*

▶ **Lemma 20** (Lemma 5.3 in [18]). *For an edge of type (i, j) located at $(u_1, u_2, \ldots, u_{k+1})$, where $i \neq j$ are indices in $[k + 1]$, 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 $\ell \in [k] \setminus \{i, j\}$.*
- $(1 - \int_{u_i}^{u_j} \psi(u)du)((1 - \int_{u_i}^{u_\ell} \psi(u)du))\psi(u_i) + \psi(u_j)$, *if $u_i < u_\ell \leq u_j$ for some other $\ell \in [k] \setminus \{i, j\}$.*
- $(1 - \int_{u_i}^{u_j} \psi(u)du)(1 - \int_{u_i}^{u_\ell} \psi(u)du)\psi(u_i) + (1 - \int_{u_j}^{u_\ell} \psi(u)du)\psi(u_j)$, *if $u_i \leq u_j < u_\ell$ for some other $\ell \in [k] \setminus \{i, j\}$.*

The proof for each of these Lemmas is exactly as in [18], save for one additional trivial observation: the cut density of an edge of type $(i, k + 1)$ is at most that of an edge of type (i, j) for any $j \neq i, j \neq k + 1$. This is because the $(k + 1)$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 [18] shows, with a specific choice of parameters, that Algorithm 7 is a 1.2965-approximation to Lifted Cut as well.