Improved Approximation Algorithm for Requirement Cut

Anupam Gupta* Viswanath Nagarajan† R. Ravi‡

Abstract

This note presents improved approximation guarantees for the *requirement cut* problem: given an n-vertex edge-weighted graph $G = (V, E)$, g groups of vertices $X_1, \cdots, X_g \subseteq V$ with each group X_i having a requirement r_i between 0 and $|X_i|$, the goal is to find a minimum cost set of edges whose removal separates each group X_i into at least r_i disconnected components. We give a tight $\Theta(\log g)$ approximation ratio for this problem when the underlying graph is a tree, and show how this implies an $O(\log k \cdot \log g)$ approximation ratio for general graphs, where $k = |\cup_{i=1}^{g} X_i| \leq n$.

Keywords: Graph Partitioning, Approximation Algorithms.

1 Introduction

We study the requirement cut problem [8] which is a generalization of several known graph partitioning problems. The input to *requirement cut* is an n-vertex undirected graph $G = (V, E)$ with edge-costs $c : E \to \mathbb{R}_+$, and g groups of vertices $X_1, \cdots, X_g \subseteq V$ with each group X_i having an integer valued requirement r_i between 0 and $|X_i|$. The objective is to find a set $S \subseteq E$ of edges minimizing the cost $\sum_{e \in S} c_e$, such that each group X_i (for $i \in [g]$) lies in at least r_i disconnected components of $G \setminus S$. We denote $k := |\cup_{i=1}^{g} X_i| \leq n$ to be the number of vertices that participate in at least one group. Requirement cut generalizes several previously studied cut problems such as *Multicut* [6], *Multiway cut* [3], *Multi-Multiway cut* [1], *Steiner Multi cut* [5], and *k-cut* [9].

In this note, we obtain better approximation ratios for the requirement cut problem. When the graph G is a tree, we obtain a tight $\Theta(\log g)$ approximation guarantee (Section 2). This improves over the $O(\log(gR))$ bound from [8], where $R = \max_{i=1}^{g} r_i$ is the maximum requirement. For instances of requirement cut on general graphs, we give an $O(\log k \cdot \log g)$ approximation algorithm (Section 3) improving over $O(\log n \cdot \log(gR))$ from [8]. The *Steiner Multicut* problem [5] is the special case of requirement cut when all requirements are two. Our $O(\log k \cdot \log g)$ approximation ratio represents a logarithmic improvement (in some cases) over the previously best-known bounds of $O(\log^3 gt)$ [5] and $O(\log n \cdot \log g)$ [8] (here $t := \max_{i=1}^{g} |X_i|$ is the maximum group size; note that $k \leq \min\{n, gt\}$).

All our algorithms are based on rounding a natural LP relaxation [8] for requirement cut. Our improvement in the tree case comes from a better rounding algorithm that first randomly partitions the tree (in a dependent manner) into parts of small diameter, and then applies randomized

[*] Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213.

[†] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA. Email: viswanath@us.ibm.com.

[‡] Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

rounding as in [8]. The improvement for general graphs relies on a slightly stronger statement of the FRT [4] tree embedding (Theorem 7).

We note that in recent work [7] also obtained an approximation guarantee of $O(\log^{4.5} k \, \log(gR))$ for requirement cut, that is independent of the graph size n. However the focus in that paper was more general, namely constructing vertex-sparsifiers that approximate all *terminal min-cuts*, and their requirement cut result follows as an application. On the other hand, in this paper, we deal directly with the requirement cut problem and obtain improved guarantees.

1.1 Linear programming relaxation

The following is a natural LP relaxation for requirement cut with a variable $z_{\{u,v\}}$ for each *unordered* pair $u, v \in V$ of vertices. This is also the linear program that was studied in [8]. A similar LP for the *Steiner k-cut* problem was studied in [2].

$$
\begin{array}{lll}
 & \min & \sum_{e\in E} c_e z_e \\
(\mathcal{LP}) & s.t. & \sum_{e\in T_i} z_e \geq r_i - 1 \qquad \forall\, T_i : \text{spanning tree in complete graph on } X_i, \\
 & & \qquad\qquad\qquad\qquad\quad \forall\, i = 1 \cdots g \\
 & & z_{\{u,w\}} \leq z_{\{u,v\}} + z_{\{v,w\}} \quad \forall\, u, v, w \in V \\
 & & 0 \leq z_{\{u,v\}} \leq 1 \qquad\qquad\quad \forall\, u, v \in V
\end{array}
$$

Note that the second constraint (triangle inequality) ensures that z defines a metric on vertices V. Also in the first constraint, T_i ranges over all spanning trees in the complete graph $\left(X_i, \binom{X_i}{2}\right)$; although some edges of T_i may be absent in E, their lengths under z are well-defined. The linear program $\mathcal{LP}$ can be solved in polynomial time using the ellipsoid algorithm. Let d^* denote an optimal solution to $\mathcal{LP}$. Our rounding algorithms work with a slightly different length function d, defined as $d_{\{u,v\}} := \min\{2 \cdot d^*_{\{u,v\}}, 1\}$ for all $u, v \in V$. Since d^* is a metric, so is d. It is also clear that all lengths in both d^* and d are in $[0, 1]$. We state a useful claim about metric d, which is Claim 1 from [8].

Claim 1 ([8]) *For any group X_i ($i = 1, \cdots g$), the minimum Steiner tree on X_i under metric d has length at least $r_i - 1$.*

2 Requirement Cut on trees

In this section, we consider instances of requirement cut when the underlying graph is a tree $T = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$. The algorithm is based on rounding the fractional solution d to $\mathcal{LP}$ (see Section 1.1). Applying randomized rounding on this fractional solution, [8] obtained an $O(\log(gR))$ approximation guarantee. We give a different rounding scheme that achieves a better $O(\log g)$ bound. Our rounding scheme consists of two stages: first we randomly partition the tree T into several subtrees of small diameter, and then apply randomized rounding on each subtree.

The first stage of rounding pays an initial logarithmic factor in the cost but ensures that the diameters of the resulting subtrees are at most inverse logarithmic. The second stage now proceeds differently from [8] in that rather than rounding edges according to their LP-value in a logarithmic number of rounds to ensure feasibility, we instead boost the rounding probability by a logarithmic factor and perform only one stage of rounding. The small diameter is crucial in allowing the probability boosting in this single stage rounding step and this is where the first stage preprocessing is useful. The algorithm is formally described in Figure 1, and its analysis is given below.

Input: An instance of requirement cut on tree $T = (V, E)$.

1. Solve the LP relaxation $\mathcal{LP}$ and obtain solution d as described in Section 1.1.
2. Set $\alpha := \frac{1}{64(\log g+1)}$, and choose $\eta \in [0, \alpha]$ uniformly at random. Let $w \in V$ be an arbitrary vertex designated as *root*.
3. For $u, v \in V$, let $l(u, v)$ denote the sum of d values on edges of the unique $u - v$ path in T.
4. **First stage:** Pick into E_1, all edges of E that are at l-distance $p \cdot \alpha + \eta$ from w (for *any* integer $p \geq 0$): i.e. $E_1 \leftarrow \{(u, v) \in E \mid l(w, u) < p \cdot \alpha + \eta \leq l(w, v), \text{ for some } p \in \mathbb{Z}_+\}$.
5. Let $\mathcal{T} := \{T_j\}_{j=1}^{q}$ denote the subtrees of T in $T \setminus E_1$.
6. **Second stage:** Pick into E_2, each edge e of $E \setminus E_1$ with independent probability $d'_e := \frac{d_e}{2\alpha}$.

Output: $E_1 \cup E_2$ as solution to the requirement cut instance.

Figure 1: Algorithm for requirement cut on trees.

First stage rounding. Note that by triangle inequality, $d_{\{u,v\}} \leq \min\{l_{\{u,v\}}, 1\}$ for all $u, v \in V$. The first stage (Step (4)) cuts edges $E_1 \subseteq E$, and let $C_1 = \sum_{e \in E_1} c_e$ its cost. Observe that the probability that an edge $e \in E$ is cut in Step (4), $\Pr[e \in E_1] = \min\{\frac{d_e}{\alpha}, 1\}$. Thus the expected cost of edges cut in this stage $E[C_1] \leq \frac{1}{\alpha} \sum_{e \in E} c_e \cdot d_e$.

Note that the above cutting procedure ensures that the diameter (under metric d) of each subtree is at most 2α. For each group X_i (where $i \in [g]$), let s_i denote the number of distinct subtrees among $\mathcal{T}$ that contain terminals from X_i, and let $r'_i := \max\{r_i - s_i, 0\}$ be the residual requirement of group X_i. By renumbering groups, let $X_1, \cdots, X_{g'}$ (for $g' \leq g$) denote the groups with positive residual requirement. In the second stage of rounding, it suffices to restrict attention to the groups $\{X_i\}_{i=1}^{g'}$.

Second stage rounding. The analysis of this part is *conditioned* on any instantiation of forest $\mathcal{T}$ that results after the first stage. It is clear that each edge $e \in E \setminus E_1$ has $d_e \leq \alpha$. The edges picked in the second stage (Step (6)) are $E_2 \subseteq E \setminus E_1$, and let $C_2 := \sum_{e \in E_2} c_e$ be its cost. Since each edge $e \in E$ is included in E_2 with probability at most $\frac{d_e}{2\alpha}$, the expected cost $E[C_2] \leq \frac{1}{2\alpha} \sum_{e \in E} c_e \cdot d_e$. We will show that E_2 satisfies *all* residual requirements with high probability. This would imply that $E_1 \cup E_2$ is a feasible integral solution to the requirement cut instance w.h.p.

Before we prove the main lemma about feasibility of the obtained solution, we introduce some definition. Let $d' : \binom{V}{2} \to \mathbb{R}_+$ be a new distance function on V defined as follows.

$$d'_{\{u,v\}} := \begin{cases} 1 & \text{if } u \text{ and } v \text{ lie in different subtrees of } \mathcal{T} \\ \frac{d_{\{u,v\}}}{2\alpha} & \text{if } u \text{ and } v \text{ lie in the same subtree of } \mathcal{T} \end{cases} \qquad \forall u, v \in V$$

Observe that for any vertices u, v in the same subtree of $\mathcal{T}$, we have $d_{\{u,v\}} \leq 2\alpha$ (this follows from the first stage rounding). So d' takes values in $[0, 1]$. It also follows that d' satisfies triangle inequality, and hence it is a metric. Note that Step (6) picks each edge e of $\mathcal{T}$ independently with probability d'_e. We now prove the following key property of the second stage rounding.

Lemma 2 *For any group $i \in [g']$, the probability that X_i lies in fewer than r_i distinct components of $\mathcal{T} \setminus E_2$ is at most $\frac{1}{4g^2}$.*

Proof: We first state some definitions and claims proved in [8] that are useful in this proof. Vertices in $V \setminus X_i$ are called *Steiner*, and X_i-vertices are called *terminal.* Define H_i as the forest obtained from $\mathcal{T}$ by repeatedly removing all degree one Steiner vertices and short-cutting over all degree two Steiner vertices. This ensures that all Steiner vertices in H_i have degree at least three. Note that H_i is not necessarily a subgraph of $\mathcal{T}$; however each edge of H_i corresponds to a path in $\mathcal{T}$, and the paths for different edges of H_i are disjoint. We say that an edge $e \in H_i$ is removed/disconnected iff some edge in the path corresponding to e is removed in $\mathcal{T}$ (in the second stage). The following claim is Claim 2 proved in [8].

Claim 3 ([8]) *The removal of any $m \geq 1$ edges of forest H_i results in at least $\lceil \frac{m+1}{2} \rceil$ more components containing terminals.*

In the following, for any subset $A \subseteq \binom{V}{2}$ of edges and metric z on V, $z(A) := \sum_{e \in A} z_e$ is the length of A under z. For any pair of vertices $u, v \in V$ that lie in the same subtree of $\mathcal{T}$, let $p_{u,v}$ denote the *probability* that vertices u and v are disconnected in the second stage rounding; i.e. $p_{u,v}$ is the probability that some edge on the $u - v$ path in forest $\mathcal{T}$ is picked into E_2. Consider a 0-1 random variable Z^i_e for each edge $e = (u, v) \in H_i$ which is 1 iff vertices u and v are disconnected, and 0 otherwise; note that $Pr[Z^i_e = 1] = p_e$. Since the edge-sets in forest $\mathcal{T}$ corresponding to edges $e \in H_i$ are disjoint, the random variables Z^i_e (for $e \in H_i$) are independent. Let $Y_i = \sum_{e \in H_i} Z^i_e$, denote the number of edges of H_i that are disconnected in the second stage rounding. Claim 3 implies that the increase in the number of X_i-components is at least $Y_i/2$. The next claim is Lemma 1 from [8].

Claim 4 ([8]) *$E[Y_i] = \sum_{e \in H_i} p_e \geq (1 - \frac{1}{e}) \cdot d'(H_i)$.*

Since Y_i is a sum of independent 0-1 random variables, a Chernoff bound as in [8] gives:

$$\Pr\left[\text{increase in } X_i\text{-components less than } \frac{1}{4}E[Y_i]\right] \leq \Pr\left[Y_i < \frac{1}{2}E[Y_i]\right] \leq e^{-E[Y_i]/8} \leq e^{-d'(H_i)/16} \tag{1}$$

where the first inequality is Claim 3 and the last inequality follows from Claim 4.

Claim 5 *$d'(H_i) \geq \frac{1}{2\alpha} r'_i$.*

Proof: Note that forest H_i consists of exactly s_i components. Consider a tree $\tilde{H}_i$ obtained by adding $s_i - 1$ edges connecting the s_i subtrees of H_i. Clearly $\tilde{H}_i$ is a Steiner tree on X_i, so Claim 1 implies $d(\tilde{H}_i) \geq r_i - 1$. Since each edge has d-length at most 1, we obtain $d(H_i) \geq d(\tilde{H}_i) - (s_i - 1) \geq r'_i$. By the definition of metric d', we have $d'(H_i) = \frac{1}{2\alpha} d(H_i) \geq \frac{1}{2\alpha} r'_i$. ■

Using Claim 4 with the above, $E[Y_i] \geq \frac{1}{4\alpha} r'_i \geq 4 \cdot r'_i$. Using inequality (1),

$$\begin{aligned}\Pr\left[\text{increase in } X_i\text{-components less than } r'_i\right] &\leq \Pr\left[\text{increase in } X_i\text{-components less than } \frac{1}{4}E[Y_i]\right] \\ &\leq e^{-d'(H_i)/16} \leq e^{-r'_i/(32\alpha)} \leq e^{-2\log g - 2} \leq \frac{1}{4g^2}\end{aligned}$$

The third to last inequality uses Claim 5, and the second to last inequality uses $r_i' \geq 1$ and $\alpha = \frac{1}{64(\log g+1)}$. This proves the lemma since X_i was already in $s_i = r_i - r_i'$ distinct components of $\mathcal{T}$. ■

It is now easy to prove the performance guarantee of our algorithm. The probability that the resulting solution $E_1 \cup E_2$ is infeasible is at most $g' \cdot \frac{1}{4g^2} \leq \frac{1}{4g} \leq \frac{1}{4}$ by a union bound and Lemma 2. As observed earlier, the expected cost of the solution $E[C_1] + E[C_2] \leq \frac{3}{2\alpha} \sum_{e\in E} c_e \cdot d_e$. Hence the probability that the solution costs more than $\frac{6}{\alpha} \sum_{e\in E} c_e \cdot d_e$ is at most $\frac{1}{4}$. Again by a union bound, with probability at least half, $E_1 \cup E_2$ is a feasible solution of cost at most $\frac{6}{\alpha} \cdot \sum_{e\in E} c_e \cdot d_e$.

Theorem 6 *There is an LP rounding algorithm for requirement cut on trees that returns a solution of cost $O(\log g) \cdot \mathsf{LP}_{\mathsf{opt}}$ with probability at least $\frac{1}{2}$, where $\mathsf{LP}_{\mathsf{opt}}$ is the optimal LP value. Here g is the number of groups.*

This result is best possible up to constant factors since the problem is $\Omega(\log g)$ hard to approximate even on an unweighted star [8].

3 General graphs

In this section we show that the requirement cut problem on general graphs admits an $O(\log k \cdot \log g)$ approximation algorithm. The main ingredient to this result is the following (slight) generalization of the FRT tree embedding [4]. A *tree metric* on vertices V is given by a tuple (T, κ) where T is a tree on vertex-set $U \supseteq V$ and $\kappa : \binom{U}{2} \to \mathbb{R}_+$ is a metric on U satisfying for all $x, y \in U$, $\kappa_{\{x,y\}} = \sum_{e\in P_{xy}} \kappa_e$ where P_{xy} denotes edges on the unique $x - y$ path in T. Note that although we are only interested in distances between vertices of V, we allow the tree to be defined on a larger vertex-set than V.

Theorem 7 *Suppose d is a metric on a finite vertex-set V with a designated subset $W \subseteq V$. Then there is a distribution $\mathcal{T}$ of tree metrics on V satisfying:*

1. **Non-contracting.** *For all $(T, \kappa) \in \mathcal{T}$, $\kappa_{\{x,y\}} \geq d_{\{x,y\}}$ for all $x, y \in W$.*
2. **Low stretch.** *$E_{\mathcal{T}}[\kappa_{\{u,v\}}] \leq \rho \cdot d_{\{u,v\}}$, for all $u, v \in V$, where $\rho = O(\log |W|)$.*

Furthermore, trees from this distribution can be sampled in polynomial time.

Proof: The proof of this theorem is essentially identical to that in [4] (where $W = V$). We only state the modification required in the algorithm, and how the analysis follows from [4]. We will describe an algorithm that generates a random tree metric from the desired distribution $\mathcal{T}$.

We assume without loss of generality that the smallest distance in d is 1. Choose $\delta \in \mathbb{N}$ such that $2^{\delta-1} \leq \max_{u,v\in V} d_{\{u,v\}} < 2^\delta$. A *cluster* refers to any subset of vertices V, and a *partial partition* is a collection of disjoint clusters. Extending a definition from [4], we define a *hierarchical cut decomposition* of (V, d) as a sequence $D_0, \cdots, D_\delta$ of partial partitions such that $D_\delta = \{V\}$ and:

- For each $0 \leq i < \delta$, D_i is a partition of some subset of clusters from D_{i+1}.
- Each cluster in D_i has radius at most 2^i, for all $0 \leq i \leq \delta$.

Hierarchical cut decomposition to tree metric. Any hierarchical cut decomposition of (V, d) naturally corresponds to a rooted tree T' as follows. Each cluster in $\cup_{i=0}^{\delta} D_i$ corresponds to a vertex in T', and the vertex corresponding to cluster V (in D_δ) is denoted root. The children in T' (if any) of cluster $S \in D_{i+1}$ are those clusters of D_i that are subsets of S. We also define a length function l on edges of T', where any edge between a D_{i+1}-cluster and a D_i-cluster has length 2^i. We now extend tree T' to obtain another tree T that has vertices V as its leaves: each leaf vertex of T' (corresponding to say cluster S) has $|S|$ new children in T that are labeled $\{v \mid v \in S\}$. The new edges that are introduced in T all have zero length (under function l). Let U denote the vertices of the resulting tree T: note that the internal vertices of T correspond to clusters in $\cup_{i=0}^{\delta} D_i$, and the leaf vertices of T are precisely V. Finally we define metric $\kappa : \binom{U}{2} \to \mathbb{R}_+$ where for any $x, y \in U$, $\kappa_{\{x,y\}} := \sum_{e \in P_{xy}} l(e)$ (here P_{xy} denotes edges on the unique $x-y$ path in T). Thus any hierarchical cut decomposition of (V, d) corresponds to a tree metric (T, κ) on V.

The algorithm. We now present a modified version of the algorithm from [4] that creates the desired hierarchical cut decomposition (c.f. algorithm $Partition(V, d)$ [4]). There are two differences from [4]: we take a permutation on W (instead of a permutation on the entire vertex set V), and a cluster's decomposition stops when it has at most one W-vertex (as opposed to stopping when it has at most one V-vertex).

1. Choose u.a.r. permutation π of W.
2. Choose β u.a.r. from $[1, 2]$.
3. Set $\delta \leftarrow \lceil \log_2 \left(\max_{u,v \in V} d_{\{u,v\}} \right) \rceil$, $D_\delta \leftarrow \{V\}$, $i \leftarrow \delta - 1$.
4. While D_{i+1} has any cluster containing more than one W-vertex:
   (a) For each cluster S in D_{i+1} containing more than one W-vertex, do:
      - Define all vertices in S to be unassigned.
      - For $l = 1, \cdots, |W|$ do: assign all unassigned vertices from $\{v \in S \mid d_{\{\pi(l),v\}} \leq \beta \cdot 2^{i-1}\}$ to a new cluster in D_i.
      - Assign all remaining unassigned vertices of S to a new cluster in D_i.

   (b) $i \leftarrow i - 1$.

The non-contracting condition in the theorem follows immediately from this construction. For every pair $x, y \in W$, the termination condition implies an $0 \leq i < \delta$ such that x and y are in different clusters of D_i, and in the same cluster of D_{i+1}. Hence $\kappa_{\{x,y\}} \geq 2^i + 2^i = 2^{i+1}$ and $d_{\{x,y\}} \leq 2^{i+1}$ (diameter bound on D_{i+1}-clusters), i.e. $\kappa_{\{x,y\}} \geq d_{\{x,y\}}$.

The low stretch condition follows directly from the analysis in [4]. ■

The algorithm for requirement cut is identical to that of [8]: the only difference is the use of this stronger tree embedding in Step (2) below. Given an instance $\mathcal{I} = \langle V, E, c, \{X_i, r_i\}_{i=1}^{g} \rangle$ of requirement cut, this proceeds as in Figure 2.

The non-contraction condition in Theorem 7 with $W = \cup_{i=1}^{g} X_i$ suffices to show (as in [8]) that κ' is a feasible fractional solution to $\mathcal{LP}(\mathcal{J})$. Namely, for any $x, y \in W$, we have $d^*_{\{x,y\}} \leq \kappa_{\{x,y\}}$ and $d^*_{\{x,y\}} \leq 1$; so $\kappa'_{\{x,y\}} = \min\{\kappa_{\{u,v\}}, 1\} \geq d^*_{\{x,y\}}$. Hence for any spanning tree T_i on group X_i, T_i's length under κ' is at least as much as its length under d^* (since $X_i \subseteq W$), which is at least $r_i - 1$ by the feasibility of d^* in $\mathcal{LP}(\mathcal{I})$. So κ' is a feasible fractional solution to $\mathcal{LP}(\mathcal{J})$.

```
Input: An instance $\mathcal{I}$ of requirement cut.
  1. Solve the LP relaxation $\mathcal{LP}(\mathcal{I})$ to obtain metric $(V, d^*)$.
  2. Apply Theorem 7 on $(V, d^*)$ with $W = \cup_{i=1}^{g} X_i$ to obtain a random tree metric $(T, \kappa)$ on $V$. Let $V'$ and $E'$ denote the vertex-set and edge-set of $T$.
  3. Define the cost of each edge $e \in E'$ as $c'_e := \sum_{f \in \mathsf{Sep}_e} c_f$, where $\mathsf{Sep}_e = \{(i,j) \in E \mid e \text{ lies on the } i - j \text{ path in } T\}$. Define $\kappa'_{\{u,v\}} := \min\{\kappa_{\{u,v\}}, 1\}$ for all $u, v \in V'$.
  4. Consider an instance $\mathcal{J}$ of requirement cut on tree $T$ with costs $c'$ and requirements $\{X_i, r_i\}_{i=1}^{g}$.
  5. Treating $\kappa'$ as a fractional solution to linear program $\mathcal{LP}(\mathcal{J})$, round it using Theorem 6 to an integral solution $S \subseteq E'$.
Output: the edges $\cup_{e \in S} \mathsf{Sep}_e$ as the solution to $\mathcal{I}$.
```

Figure 2: Algorithm for requirement cut on general graphs.

Exactly as in [8], the low stretch condition in Theorem 7 now implies that the expected cost (under c') of metric κ' is at most $\rho \cdot \sum_{f \in E} c_f \cdot d^*_f$. Here $\rho = O(\log k)$, for $k = |W| = |\cup_{i=1}^{g} X_i|$. Using the $O(\log g)$ LP rounding algorithm of Theorem 6 in Step (5), it follows that the cost of solution S is $O(\rho \cdot \log g) \cdot \sum_{f \in E} c_f \cdot d^*_f$. Finally it is easy to see that if S is a feasible solution to $\mathcal{J}$, then $\cup_{e \in S} \mathsf{Sep}_e$ is a feasible solution to $\mathcal{I}$, and $\sum_{e \in S} c'_e = \sum_{f \in \cup_{e \in S} \mathsf{Sep}_e} c_f$. This proves the following.

Theorem 8 *There is an LP rounding algorithm for requirement cut that returns a solution of cost $O(\log k \cdot \log g) \cdot \mathsf{LP}_{\mathsf{opt}}$ with probability at least $\frac{1}{2}$, where $\mathsf{LP}_{\mathsf{opt}}$ is the optimal LP value. Here g is the number of groups, and $k = | \cup_{i=1}^{g} X_i|$ is the number of vertices participating in the groups.*

Acknowledgement

Anupam Gupta's research was partly supported by the NSF CAREER award CCF-0448095 and an Alfred P. Sloan Fellowship. R. Ravi's research was supported in part by NSF grant CCF-0728841.

References

[1] A. Avidor and M. Langberg. The Multi-Multiway cut problem. *Theoretical Computer Science*, 377(1-3):35–42, 2007.

[2] Chandra Chekuri and Sudipto Guha. The Steiner k-cut problem. *SIAM Journal on Discrete Mathematics*, 20(1):261–271, 2006.

[3] 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.

[4] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

[5] 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.

[6] P.N. Klein, S. Rao, A. Agarwal, and R. Ravi. An approximate max-flow min-cut relation for multicommodity flow, with applications. *Combinatorica*, 15:187–202.

[7] Ankur Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *Proceedings of FOCS*, 2009.

[8] Viswanath Nagarajan and R Ravi. Approximation algorithms for requirement cut on graphs. *Algorithmica*, 56(2):198–213, 2010.

[9] Huzur Saran and Vijay V. Vazirani. Finding k Cuts within Twice the Optimal. *SIAM Journal on Computing*, 24(1):101–108, 1995.