Approximation Algorithms for Multicommodity-Type Problems with Guarantees Independent of the Graph Size

Ankur Moitra*
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
moitra@mit.edu

Abstract— Linial, London and Rabinovich [16] and Aumann and Rabani [3] proved that the min-cut max-flow ratio for general maximum concurrent flow problems (when there are k commodities) is $O(\log k)$. Here we attempt to derive a more general theory of Steiner cut and flow problems, and we prove bounds that are poly-logarithmic in k for a much broader class of multicommodity flow and cut problems. Our structural results are motivated by the meta question: Suppose we are given a $poly(\log n)$ approximation algorithm for a flow or cut problem - when can we give a $poly(\log k)$ approximation algorithm for a generalization of this problem to a Steiner cut or flow problem?

Thus we require that these approximation guarantees be independent of the size of the graph, and only depend on the number of commodities (or the number of terminal nodes in a Steiner cut problem). For many natural applications (when $k = n^{o(1)}$) this yields much stronger guarantees.

We construct vertex-sparsifiers that approximately preserve the value of all terminal min-cuts. We prove such sparsifiers exist through zero-sum games and metric geometry, and we construct such sparsifiers through oblivious routing guarantees. These results let us reduce a broad class of multicommodity-type problems to a uniform case (on k nodes) at the cost of a loss of a $poly(\log k)$ in the approximation guarantee. We then give $poly(\log k)$ approximation algorithms for a number of problems for which such results were previously unknown, such as requirement cut, l-multicut, oblivious 0-extension, and natural Steiner generalizations of oblivious routing, min-cut linear arrangement and minimum linear arrangement.

Keywords-multicommodity flow; metric geometry; approximation algorithms;

1. Introduction

Linial, London and Rabinovich [16] and Aumann and Rabani [3] proved that the min-cut max-flow ratio for general maximum concurrent flow problems (when there are k commodities) is $O(\log k)$. These results imply approximation algorithms for a variety of NP-hard cut problems and give an approximation guarantee that is poly-logarithmic in k as opposed to poly-logarithmic in n. Garg, Vazirani and Yannakakis [9] proved that the min-cut max-flow ratio for maximum multicommodity flow is also $O(\log k)$. Here we attempt to derive a more general theory of Steiner cut and flow problems, and we prove bounds that are poly-logarithmic in k. These bounds apply to a much broader class of multicommodity flow and cut problems. Our structural results are motivated by the meta question: Suppose we are given a $poly(\log n)$ approximation algorithm for a flow or cut problem - when can we give a $poly(\log k)$ approximation algorithm for a generalization of this problem to a Steiner cut or flow problem?

Thus we require that these approximation guarantees be independent of the size of the graph, and only depend on the number of commodities (or the number of terminal nodes in a Steiner cut problem). For many natural applications of multicommodity flows and cuts, we expect that the number of commodities k is much smaller than n, and for such problems we get approximation algorithms that have much stronger guarantees.

1.1. Concisely, and Simultaneously Approximating All Cuts

Suppose we are given an undirected, capacitated graph $G = (V, E)$ and a set $K \subset V$ of size k. Let $h : 2^V \rightarrow \Re^+$ denote the cut function of G:

$$h(A) = \sum_{(u,v)\in E \text{ s.t. } u\in A, v\in V-A} c(u,v)$$

We define the function $h_K : 2^K \rightarrow \Re^+$ which we refer to as the terminal cut function on K:

$$h_K(U) = \min_{A\subset V \text{ s.t. } A\cap K = U} h(A)$$

The combinatorial interpretation of the terminal cut function is that $h_K(U)$ is just the minimum edge cut separating U from $K - U$ in G. Note that U is required to be a subset of K. We prove that there is an undirected, capacitated graph $G' = (K, E')$ *on just the terminals* so that the cut function of this graph $h' : 2^K \rightarrow \Re^+$ satisfies

$$h_K(U) \leq h'(U) \leq O(\frac{\log k}{\log\log k})h_K(U)$$

for all $U \subset K$. So we can approximate the terminal cuts in G on a graph G' of size k and this approximation factor is independent of the size of the graph G. Previous results - see

*This research was supported in part by a Fannie and John Hertz Foundation Fellowship. Part of this work was done while the author was visiting Microsoft Research Redmond.

Mader's Theorem or [14] - on approximating the terminal cuts of a graph only preserve the smallest terminal cuts. Such papers approach approximating terminal cuts via matroid theory, and here we consider the problem of approximating *all* terminal cuts via metric geometry.

This (surprising) structural result is a crucial step in our paper: For almost all non-pathological Steiner multicommodity cut and flow problems, we can solve the problem in G' and map a solution back to G while losing only $poly(\log k)$ factors in the approximation guarantee.

Open Question 1: Is there an $\omega(1)$ lower bound on how well the cut function of a graph $G' = (K, E')$ can approximate the terminal cut function?

1.2. An Approach Through Metric Geometry

Our approach fundamentally relies on metric geometry and oblivious routing. We use a rounding algorithm due to [7] for the 0-extension problem to get an *existential* result on approximating the terminal cuts of a graph concisely in an undirected, capacitated graph G' on only the nodes K (a vertex sparsifier). Roughly, we prove this structural result by realizing it as the min-max dual to the 0-extension problem.

We note that in addition to the many known applications of oblivious routing schemes, here we use oblivious routing guarantees to approximate a geometric question about polytopes: Given polytopes $Q, P \subset \Re^n$ (that are given by a separation oracle), find a unit vector $u \in \Re^n$ that maximizes the ratio

$$\frac{\max \lambda_P \text{ s.t. } \lambda_P u \in P}{\max \lambda_Q \text{ s.t. } \lambda_Q u \in Q}$$

We can use Lowner-John ellipsoids to get an $O(n^3)$-approximation to this problem [11], but in the special case in which Q is the set of demand vectors that are routable in an undirected, capacitated graph G with congestion at most 1, we give a $O(\log n)$ approximation algorithm for this problem. We accomplish this by using oblivious routing schemes to give a geometric relaxation for Q. We believe that this notion of using an oblivious routing scheme to approximately solve an optimization problem can be useful in other applications.

We use this approximation algorithm to give a polynomial time construction of a graph G' on K (a vertex sparsifier) that approximates the terminal cuts of the original graph G (to within a worse but still $poly(\log k)$ factor).

These results let us reduce a broad class of multicommodity-type problems to a uniform case (on k nodes) at the cost of a loss of a $poly(\log k)$ in the approximation guarantee. We cannot concisely define this class but we can use our results to give $poly(\log k)$ approximation algorithms for a number of problems for which such results were previously unknown, such as requirement cut [1] , l-multicut, and natural Steiner generalizations of oblivious routing, min-cut linear arrangement and minimum linear arrangement.

We can also give a $poly(\log k)$ approximation algorithm for the 0-extension problem that is *oblivious* to the semi-metric Δ (defined on a fixed K). We defer the proof to the full version.

2. Maximum Concurrent Flow

An instance of the maximum concurrent flow problem consists of an undirected graph $G = (V, E)$, a capacity function $c : E \to \Re^+$ that assigns a non-negative capacity to each edge, and a set of demands $\{(s_i, t_i, d_i)\}$ where $s_i, t_i \in V$ and d_i is a non-negative real value. For such problems we set $K = \cup_i \{s_i, t_i\}$ and suppose that $|K| = k$. The maximum concurrent flow question asks, given such an instance, what is the largest fraction of the demand that can be simultaneously satisfied? This problem can be formulated as a polynomial-sized linear program, and hence can be solved in polynomial time. However, a more natural formulation of the maximum concurrent flow problem can be written using an exponential number of variables.

For any $a, b \in V$ let $P_{a,b}$ be the set of all (simple) paths from a to b in G. Then the maximum concurrent flow problem can be written as :

$$\begin{array}{ll} \max & \lambda \\ \text{s.t.} & \\ & \sum_{P \in P_{s_i,t_i}} x(P) \geq \lambda d_i \\ & \sum_{P \ni e} x(P) \leq c(e) \\ & x(P) \geq 0. \end{array}$$

For a maximum concurrent flow problem, let λ^* denote the optimum. We defined the cut function h and the terminal cut function h_K in section 1.2. We also define the demand function $d : 2^K \to \Re^+$ as

$$d(U) = \sum_{i \text{ s.t. } |\{s_i,t_i\} \cap U = 1|} d_i$$

which given $U \subset K$ is just the total demand that has exactly one endpoint in U and one endpoint in $K - U$.

Theorem 1: [16] [3] If all demands are supported in K, and $|K| = k$, then there exists a cut $A \subset V$ such that

$$\frac{h(A)}{d(A \cap K)} \leq O(\log k)\lambda^*$$

We are interested in multicommodity-type problems, which we informally define as problems that are only a function of the terminal cut function h_K and the congestion

[1] Subject to the mild technical restriction that the number of groups g be at most quasi-polynomial in k. Otherwise we give a $O(poly(\log k) \log g)$-approximation algorithm and if g is not quasi-polynomial in k then this approximation algorithm is dominated by $O(\log g)$ and there is a lower bound of $\Omega(\log g)$ for the approximability of this problem via a reduction from set cover [17].

of multicommodity flows with demands supported in the set K. We want to find a graph $G' = (K, E')$ and a capacity function $c' : E' \to \Re^+$ such that for all $U \subset K$:

$$h_K(U) \leq h'(U) \leq poly(\log k) h_K(U)$$

where $h' : K \to \Re^+$ is the cut function defined on the graph G'. For non-pathological Steiner cut problems (such as, for example the requirement cut problem), mapping solutions between G and G' will preserve the value of the solution to within a $poly(\log k)$ factor. This strategy is the basis for the approximation algorithms designed in this paper.

But for multicommodity-type problems which depend on the congestion of certain multicommodity flows, we need a method to preserve the congestion of all multicommodity flows within a $poly(\log k)$ factor. The above theorem due to Linial, London and Rabinovich and Aumann and Rabani gives a $O(\log k)$-approximate min-cut max-flow relation for maximum concurrent flows, and this theorem allows us to use reductions that approximately preserve the terminal cut function to approximately preserve the congestion of all multicommodity flows too.

Throughout we will use the notation that graphs G_1, G_2 (on the same node set) are "summed" by taking the union of their edge set (and allowing parallel edges).

3. Structural Graph Theory as a Dual to Metric Geometry

Here we prove that there is graph $G' = (K, E')$ and a capacity function $c' : E' \to \Re^+$ such that for all $U \subset K$:

$$h_K(U) \leq h'(U) \leq O(\frac{\log k}{\log \log k}) h_K(U)$$

where $h' : K \to \Re^+$ is the cut function defined on the graph G'. This is a global structural result, but we prove this by introducing a zero-sum game between an extension player and a cut player. The extension player attempts to construct such a graph G', and the cut player verifies that the cut function of this graph approximates the terminal cut function. We define this game in such a way so that bounding the game value of this game implies the above structural result.

We can bound the game value of this game by proving that there is a good response for the extension player for every distribution on checks that the cut player makes. We use a rounding procedure due to Fakcharoenphol, Harrelson, Rao and Talwar [7] for the 0-extension problem to produce a good response. The zero-sum game allows us to transform a global question about whether such graphs G' exist into a local question about finding a good response to distributions on checks.

We use a zero-sum game to realize a result in structural graph theory as the dual to a local question in metric geometry. This approach is inspired by the results of Räcke. Although not explicitly stated as such in [19], Räcke's construction can also be viewed as using a zero-sum game to derive a result in structural graph theory (the existence of a $O(\log n)$-competitive oblivious routing scheme) as the dual to the local question of low distortion embeddings into tree metrics [8].

3.1. 0-Extensions

The 0-extension problem was originally formulated by Karzanov who introduced the problem as a natural generalization of the minimum multiway cut problem [13]. Suppose we are given an undirected, capacitated graph $G = (V, E)$, $c : E \to \Re^+$, a set of terminals $K \subset V$ and a semi-metric D on the terminals. Then the goal of the 0-extension problem is to assign each node in V to a terminal in K (and each terminal $t \in K$ must be assigned to itself) such that the sum over all edges (u, v) of $c(u, v)$ times the distance between u and v under the metric D is minimized.

Formally, the goal is to find a function $f : V \to K$ (such that $f(t) = t$ for all $t \in K$) so that $\sum_{(u,v)\in E} c(u, v) D(f(u), f(v))$ is minimized over all such functions. Then when D is just the uniform metric on the terminals K, this exactly the minimum multiway cut problem. Calinescu, Karloff and Rabani gave a (semi)metric relaxation of the 0-extension problem [5]:

$$\begin{array}{ll} \min & \sum_{(u,v)\in E} c(u, v)\delta(u, v) \\ \text{s.t.} & \\ & \delta \text{ is a semi-metric on } V \\ & \forall_{t,t'\in K} \delta(t, t') = D(t, t'). \end{array}$$

Note that the semi-metric δ is defined on V while D is defined only on K. Let OPT^* denote the value of an optimal solution to the above linear programming relaxation of the 0-extension problem. Clearly $OPT^* \leq OPT$. Calinescu, Karloff and Rabani [5] gave a randomized rounding procedure to round any feasible solution δ of value C to a 0-extension that has expected value at most $O(\log k)C$. Fakcharoenphol, Harrelson, Rao and Talwar [7] gave an improved randomized rounding procedure that achieves an $O(\frac{\log k}{\log \log k})$ approximation ratio:

Theorem 2: [7]

$$OPT^* \leq OPT \leq O(\frac{\log k}{\log \log k}) OPT^*$$

Given a function $f : V \to K$ such that $f(t) = t$ for all $t \in K$, we can define the capacitated graph H on K that results from the function f as:

$$c_H(a, b) = \sum_{u,v|f(u)=a,f(v)=b} c(u, v)$$

We will abuse notation and refer to the graph H generated by f as a 0-extension of the graph G. We will use the above theorem due to Fakcharoenphol, Harrelson, Rao and Talwar to show that, existentially, there is a graph G' such that for

all $U \subset K$ $h_K(U) \leq h'(U) \leq O(\frac{\log k}{\log \log k})h_K(U)$. In fact, this graph will be a convex combination of 0-extensions of G.

3.2. A Zero-Sum Game

Here we introduce and analyze an appropriately chosen zero-sum game, so that a bound on the game value of this game will imply the desired structural graph theory result.

Given an undirected, capacitated graph $G = (V, E)$ and a set $K \subset V$ of terminals, an extension player (P1) and a cut player (P2) play the following zero-sum game that we will refer to as the extension-cut game:

The **extension player** (P1) chooses a 0-extension $f : V \to K$ such that $f(t) = t$ for all terminals t

The **cut player** (P2) chooses a cut from 2^K

Given a strategy f for P1 and a strategy A for P2, P2 wins $\frac{1}{h_K(A)}$ units for each unit of capacity crossing the cut $(A, K - A)$ in P1's 0-extension. Also, we restrict P2 to play only strategies A for which $h_K(A) \neq 0$. So if P1 plays a strategy f and P2 plays a strategy A then P2 wins:

$$N(f, A) = \sum_{(u,v)\in E} \frac{1_{|\{f(u),f(v)\}\cap A|=1} c(u, v)}{h_K(A)}$$

Definition 1: Let ν denote the game value of the extension-cut game

Using von Neumann's Min-Max Theorem, we can bound the game value by bounding the cost of P1's best response to any fixed, randomized strategy for P2. So consider any randomized strategy μ for P2. μ is just a probability distribution on 2^K. We can define an ℓ_1 metric on K:

$$D_\mu(t, t') = \sum_{A\subset K} \mu(A) \frac{1_{|\{t,t'\}\cap A|=1}}{h_K(A)}$$

D_μ is just a weighted sum of cut-metrics on K. Given D_μ we can define a semi-metric δ that is roughly consistent with D_μ. This semi-metric will serve as a feasible solution to the linear programming relaxation for the 0-extension problem. A bound on the cost of this feasible solution will imply that there is a 0-extension that has not too much cost, and this will imply that the extension player has a good response to the strategy μ. We define δ as:

Initially set all edge distances $d(u, v)$ to zero. Then for each $A \subset K$, if there is no unique minimum cut separating A and $K - A$, choose one such minimum cut arbitrarily. For this minimum cut, for each edge (u, v) crossing the cut, increment the distance $d(u, v)$ by $\frac{\mu(A)}{h_K(A)}$.

Then let δ be the semi-metric defined as the shortest path metric on G when distances are $d(u, v)$.

Claim 1: $\delta(t, t') \geq D_\mu(t, t')$ for all terminals t, t'

Claim 2: $\sum_{(u,v)\in E(G)} \delta(u, v)c(u, v) = 1$

Theorem 3:

$$\nu \leq O(\frac{\log k}{\log \log k})$$

Proof: Using the theorem due to Fakcharoenphol, Harrelson, Rao and Talwar, there exists a 0-extension $f : V \to K$ (such that $f(t) = t$ for all terminals t) and such that

$$\sum_{(u,v)\in E} c(u, v)\delta(f(u), f(v)) \leq O(\frac{\log k}{\log \log k})$$

Then suppose P1 plays such a strategy f:

$$\begin{aligned} & E_{A\leftarrow\mu}[N(f, A)] \\ &= \sum_{(u,v)\in E} \sum_{A} \frac{1_{|\{f(u),f(v)\}\cap A|=1} c(u, v)\mu(A)}{h_K(A)} \\ &= \sum_{(u,v)\in E} c(u, v)D_\mu(f(u), f(v)) \\ &\leq \sum_{(u,v)\in E} c(u, v)\delta(f(u), f(v)) \leq O(\frac{\log k}{\log \log k}) \end{aligned}$$

■

We can immediately use the bound on the game value to obtain the desired structural result:

Theorem 4: There exists a graph $G' = (K, E')$ that is a convex combination of 0-extensions of G, for which for all $A \subset K$:

$$h_K(A) \leq h'(A) \leq O(\frac{\log k}{\log \log k})h_K(A)$$

where $h' : 2^K \to \Re^+$ is the cut function on G'.

Proof: We can again apply von Neumann's Min-Max Theorem, and get that there exists a distribution γ on 0-extensions ($f : V \to K$ s.t. $f(t) = t$ for all $t \in K$) such that for all $A \subset K$:

$$E_{f\leftarrow\gamma}[N(f, A)] = O(\frac{\log k}{\log \log k})$$

For any 0-extension f, let G_f be the corresponding 0-extension of G generated by f, and let $h_f : 2^K \to \Re^+$ be the cut function defined on this graph. Then let $G' = \sum_{f\in supp(\gamma)} \gamma(f)G_f$. And for any $A \subset K$:

$$h_K(A) \leq h_f(A) \text{ and}$$

$$h_K(A) \leq h'(A) = \sum_{f\in supp(\gamma)} \gamma(f)h_f(A)$$

Also because $E_{f\leftarrow\gamma}[N(f, A)] = O(\frac{\log k}{\log \log k})$:

$$\sum_{(u,v)\in E} \sum_{f\in supp(\gamma)} \gamma(f) \frac{1_{|\{f(u),f(v)\}\cap A|=1} c(u, v)}{h_K(A)} =$$

$$\frac{1}{h_K(A)} \sum_{f\in supp(\gamma)} \gamma(f)h_f(A) = \frac{h'(A)}{h_K(A)}$$

So for any $A \subset K$

$$h'(A) \leq O(\frac{\log k}{\log \log k}) h_K(A)$$

■

Note that this proof is non-constructive. In particular, the Garg-Konemann inspired techniques that are used in [19] to give a polynomial time construction for a $O(\log n)$-competitive oblivious routing scheme cannot be used to make this result constructive because even given a graph G' that is a convex combination of 0-extension graphs, checking if $h'(U) \leq O(\frac{\log k}{\log \log k}) h_K(U)$ for all $U \subset K$ is a sparsest cut question (as we will see later).

We will eventually use this existential result to prove that a certain polytope (with exponentially many constraints) is feasible, and then use entirely different techniques to find a point in the polytope and to constructively find such a graph G' that approximates the terminal cut function to within $poly(\log k)$.

4. Applications to Oblivious Routing

Suppose we are given a capacitated, undirected graph G and a subset $K \subset V$ of size k. Suppose also that we are promised all the demands we will be asked to route will have both endpoints in K. Here we prove that for this problem there is an oblivious routing scheme that is $O(\frac{\log^3 k}{\log \log k})$-competitive. We also give a polynomial (in n and k) time algorithm for constructing such schemes. There are many previously known oblivious routing schemes, but if $k^2 << \log n$ then these schemes cannot beat the trivial $\binom{k}{2}$ competitive ratio resulting from choosing $\binom{k}{2}$ independent minimum congestion unit flows (one for each $a, b \in K$).

4.1. Existential Oblivious Routing

Let G' be a convex combination of 0-extensions of G, and suppose that for all $A \subset K$:

$$h_K(A) \leq h'(A) \leq O(\frac{\log k}{\log \log k}) h_K(A)$$

Here we consider G' to be a demand graph on the terminals K.

Lemma 1: The demands in G' can be routed in G with congestion at most $O(\frac{\log^2 k}{\log \log k})$

Proof: Let $T \subset V$ be arbitrary. Also let $A = T \cap K$. Then

$$h(T) \geq h_K(A)$$

So

$$\Omega(\frac{\log \log k}{\log k}) \leq \frac{h_K(A)}{h'(A)} \leq \frac{h(T)}{d(T)}$$

and this holds for all $T \subset V$, so the sparsest cut in G (when demands are given by G') is at least

$$\Omega(\frac{\log \log k}{\log k})$$

Using the Theorem 1 due to Linal, London and Rabinovich [16] and Aumann and Rabani [3], this implies that all the demands in G' can be routed using congestion at most

$$O(\frac{\log^2 k}{\log \log k})$$

■

We also note that G' is a better communication network than G:

Lemma 2: Any set of demands (which have support only in K) that can be routed with congestion at most C, can also be routed in G' with congestion at most C

Proof:

$$G' = \sum_{f \in supp(\gamma)} \gamma(f) G_f$$

And each G_f is a 0-extension of G, so given a flow that satisfies the demands and achieves a congestion of at most C in G, we can take a flow path decomposition of this flow. Then consider any path in the flow decomposition and suppose that this flow path carries δ units of flow. Decompose this path into subpaths that connect nodes in K and contain no nodes of K as internal nodes. For each such subpath, suppose that the subpath connects a and b in K, then add δ units of flow along the edge (a, b) in G_f. This scheme will satisfy all demands because the original flow paths in G satisfied all demands. And also, each edge in G_f will have congestion at most C because the edges in G_f are just a subset of the edges in G and each edge in G_f is assigned exactly the same total amount of flow as it is in the flow in G.

So for each $f \in supp(\gamma)$, route $\gamma(f)$ fraction of all demands according to the routing scheme given for G_f above. The contribution to the congestion of any edge $(u, v) \in E'$ from any f is at most $\gamma(f)C$, and so the total congestion on any edge is at most C. Yet all demands are met because $\sum_{f \in supp(\gamma)} \gamma(f) = 1$. ■

We can now construct an oblivious routing scheme in G' and compose this with the embedding of G' into G to get an oblivious Steiner routing scheme in G:

Theorem 5: [19] There is an oblivious routing scheme for G' (on k nodes) that on any set of demands incurs congestion at most $O(\log k)$ times the off-line optimum

Theorem 6: There is an oblivious Steiner routing scheme that on any set of demands (supported in K) incurs congestion at most $O(\frac{\log^3 k}{\log \log k})$ times the off-line optimum

Proof: Given G', use Räcke's Theorem [19] to construct an oblivious routing scheme in G'. This can be mapped to an oblivious routing scheme in G using the existence of a low-congestion routing for the demand graph G' in G: Given $a, b \in K$, if the oblivious routing scheme

in G' assigns δ units of flow to a path $P_{a,b}$ in G', then construct a set of paths in G that in total carry δ units of flow as follows:

Let $P_{a,b} = (a, p_1), (p_1, p_2), ...(p_l, b)$. Let $p_0 = a$ and $p_{l+1} = b$. Then consider an edge (p_i, p_{i+1}) contained in this path and suppose that $c'(p_i, p_{i+1})$ is α in G'. Then for each flow path P connecting p_i to p_{i+1} in the low-congestion routing of G' in G, add the same path and multiply the weight by $\frac{\delta}{\alpha}$. The union of these flow paths sends δ units of flow from a to b in G. Räcke's oblivious routing scheme sends one unit of flow from a to b for all $a, b \in K$ in G'. So this implies that we have constructed a set of flows in G such that for all $a, b \in K$, one unit of flow is sent from a to b in G.

So consider any set of demands that have support contained in K. Suppose that this set of demands can be routed in G with congestion C. Then there exists a flow satisfying these demands that can be routed in G' with congestion at most C using Lemma 2. Räcke's oblivious routing guarantees imply that the oblivious routing scheme in G' incurs congestion at most $O(\log k)C$ on any edge in G'. This implies that we have scaled up each edge in G' by at most $O(\log k)C$ and so we have scaled up the amount of flow transported on each path in an optimal routing the (demand) graph G' into G by at most $O(\log k)C$. So the congestion incurred by this oblivious routing scheme is at most

$$O(\frac{\log^3 k}{\log \log k})C$$

■

This result is non-constructive, because the proof of Theorem 4 is non-constructive.

4.2. Constructive Oblivious Routing

Azar et al formulate the problem of deciding (for a given graph G) whether there exists an oblivious routing scheme that is T-competitive against the off-line optimal algorithm as a linear program [4]. This algorithm can be adapted to yield:

Theorem 7: An optimal oblivious Steiner routing scheme can be constructed in polynomial (in n and k) time

So an $O(\frac{\log^3 k}{\log \log k})$-competitive oblivious Steiner routing scheme can be constructed in polynomial time.

5. Constructions for Vertex Sparsifiers

Here we consider the problem of constructing - in time polynomial in n and k - an undirected, capacitated graph $G' = (K, E')$ for which the cut function h' approximates the terminal cut function h_K. We will use existential results (in Theorem 4) to conclude that a particular polytope is non-empty, and we will design approximate separation oracles for this polytope to give a polynomial time construction for finding such a graph G'.

Rather surprisingly, we use oblivious routing guarantees to design an approximate separation oracle. So apart from the original motivation for studying oblivious routing schemes, we actually use oblivious routing *to solve an optimization problem*. These ideas lead us to believe that the remarkable oblivious routing guarantees due to Räcke [19] can also be understood as a geometric phenomenon particular to undirected multicommodity polytopes.

5.1. The Terminal Cut Polytope

Constructing such a graph G' can be naturally represented as a feasibility question for a linear program. We can define a non-negative variable $x_{a,b}$ for each pair $a, b \in K$. Then finding a G' for which the cut function h' $f(k)g(k)$-approximates the terminal cut function is equivalent to finding a feasible point in the polytope:

$$\textbf{Type 1:} \quad \sum_{a,b \text{ s.t. } a \in A, b \notin A} x_{a,b} \leq f(k) h_K(A)$$

$$\textbf{Type 2:} \quad h_K(A) \leq g(k) \sum_{a,b \text{ s.t. } a \in A, b \notin A} x_{a,b}$$

$$0 \leq x_{a,b}$$

Theorem 4 implies that this polytope is non-empty for $f(k) = O(\frac{\log k}{\log \log k})$, $g(k) = 1$. However there are 2^{k+1} linear constraints, and we cannot check all constraints in time polynomial in n and k. We will construct approximate separation oracles for both Type 1 and Type 2 Inequalities. Then we can use the ellipsoid algorithm to find feasible edge weights. And we will choose $f(k)$ and $g(k)$ to be polylogarithmic in k.

Lemma 3: There is a polynomial time algorithm to find a Type 1 Inequality that is within an $O(\sqrt{\log k} \log \log k)$ factor approximately the maximally violated Type 1 Inequality.

Proof: Given non-negative edge weights $x_{a,b}$ we can consider the problem of (approximately) minimizing

$$\frac{h_K(A)}{h'(A)}$$

over all sets $A \subset K$. This is exactly the sparsest cut problem when the graph G' (with edge weights $x_{a,b}$) is considered to be the demand graph and we are attempting to route this demand with low congestion in G. Then we can use the current best approximation algorithm to sparsest cut due to [2] which is a $O(\sqrt{\log k} \log \log k)$ approximation algorithm to this problem, and we will find a set B for which

$$\frac{h_K(B)}{h'(B)} \leq O(\sqrt{\log k} \log \log k) \min_{A \subset K} \frac{h_K(A)}{h'(A)}$$

And so the Type 1 Inequality for the set B is within an $O(\sqrt{\log k} \log \log k)$ factor approximately the maximally

violated Type 1 Inequality, and we can find such a set constructively (with high probability). ■

We will use (constructive) algorithms for oblivious routing to find an approximately maximally violated Type 2 Inequality: Suppose there is a Type 2 constraint that is violated by a OPT-factor. So there exists a terminal cut A such that $h_K(A) > OPTh'(A)$.

Lemma 4: There exists a maximum concurrent flow $\vec{f}$ that can be routed with congestion 1 in G, but cannot be routed with congestion $\leq OPT$ in G'.

Proof: Place a super-source s and connect s via infinite capacity (directed) edges to each node in A. Also place a super-sink t and connect each node in $K-A$ via an infinite capacity (directed) edge to t. Compute a maximum $s-t$ flow. The value of this flow is $h_K(A)$. So choose a maximum concurrent flow problem $\vec{f}$ in which $f_{a,b}$ is just the amount of a to b flow in a path decomposition of the above maximum flow. In particular, $f_{a,b} = 0$ if a and b are either both in A or both in $K-A$. This flow can clearly be routed in G with congestion at most 1, because we constructed this demand vector from such a routing.

However because $\frac{h'(A)}{h_K(A)} < \frac{1}{OPT}$ there is a cut of sparsity $\frac{1}{OPT}$ and so $\vec{f}$ cannot be routed with congestion $\leq OPT$ in G'. ■

Hence we consider the problem of finding a demand vector $\vec{f}$ that can be routed with congestion 1 in G, but cannot be routed with congestion $\leq O(\log k)g(k)$ in G'. Given such a demand vector $\vec{f}$, we can find a cut of sparsity at most $< \frac{1}{g(k)}$ in G' i.e. we can find a cut $A \subset K$ for which

$$\frac{h'(A)}{d(A)} < \frac{1}{g(k)}$$

Because $\vec{f}$ can be routed with congestion 1 in G, we are guaranteed:

$$\frac{h_K(A)}{d(A)} \geq 1$$

So this implies that we have found a Type 2 Inequality that is violated. So using Lemma 4 and the above argument, up to a $O(\log k)$ factor, the problem of finding an (approximately) maximally violated Type 2 Inequality is equivalent to finding a demand vector $\vec{f}$ that can be routed with congestion at most 1 in G, and maximizes the minimum congestion needed to route this demand vector in G'. We define this problem formally as the *Max-Min Congestion Problem*, and we given a $poly(\log k)$ approximation algorithm for this problem:

Formally, given an undirected, capacitated graph $G = (V, E)$, a subset $K \subset V$ of size k, and an undirected, capacitated graph $G' = (K, E')$ the goal of the Max-Min Congestion Problem is to find a demand vector $\vec{f}$ (such that the demands are supported in K) that can be routed with congestion 1 in G and maximizes the minimum congestion needed to route $\vec{f}$ in G'.

5.2. An Approximation Algorithm via Oblivious Routing

Theorem 8: There exists a polynomial time $O(\log k)$-approximation algorithm for the Max-Min Congestion Problem.

We first construct an oblivious routing scheme for G'. Let $f'^{a,b} : E' \to \Re^+$ be an $O(\log k)$-competitive oblivious routing scheme for G' - i.e. for each $a, b \in K$, $f'^{a,b}$ sends a unit flow from a to b in G'. Then f' has the property that for any demand vector $\vec{d}$, the congestion that results from routing according to f' is within an $O(\log k)$ factor of the optimal congestion for routing $\vec{d}$ in G'. And such an oblivious routing scheme is guaranteed to exist, and can be found in polynomial time using the results due to Räcke [19].

For any edge $(a, b) \in E'$, we consider the following linear program $LP(a, b)$:

$$\begin{aligned} \max D_{a,b} \quad & = \frac{\sum_{i,j} d_{i,j} f'^{i,j}(a,b)}{c(a,b)} \\ \text{s.t.} \quad & \\ & \textstyle\sum_t x^{i,j}(i,t) = d_{i,j} \\ & \textstyle\sum_t x^{i,j}(s,t) = 0 \\ & \textstyle\sum_{i,j} x^{i,j}(e) \leq c(e) \\ & x^{i,j}(e) \geq 0. \end{aligned}$$

We will solve the above linear program for all $(a, b) \in E'$ and output the demand vector $\vec{d}$ that achieves the maximum $D_{a,b}$ over all $(a, b) \in E'$. Let

$$D = \max_{(a,b) \in E'} D_{a,b}$$

Lemma 5: Let $\vec{d}$ be the output, then $\vec{d}$ can be routed with congestion at most 1 in G and cannot be routed with congestion $< \frac{D}{O(\log k)}$ in G'.

Proof: The linear program enforces that $\vec{d}$ can be routed in G with congestion at most 1. Suppose $\vec{d}$ achieves value D on an edge (i, j) -i.e. $(i, j) = \arg\max_{(a,b)} D_{a,b}$ and $\vec{d}$ is the optimizing demand. Then $\vec{d}$ achieves congestion D on edge (i, j) when routed according to the oblivious routing scheme. The oblivious routing guarantees for G' imply that no routing of $\vec{d}$ achieves congestion smaller than

$$\frac{D}{O(\log k)}$$

■

Let OPT be the optimal value for the Max-Min Congestion Problem.

Lemma 6: There exists a feasible demand $\vec{d}$ which achieves $D_{a,b} \geq OPT$ for some $(a, b) \in E'$.

Proof: Let $\vec{d'}$ be demand that achieves the optimal value for the Max-Min Congestion Problem. The oblivious routing scheme is a routing, and must then achieve congestion at least OPT on some edge $(a,b) \in E'$ (not necessarily the same edge that achieves congestion OPT as in the optimal off-line routing scheme for $\vec{d'}$ in G'). ■

Proof: This implies Theorem 8, and the vector $\vec{d}$ computed by solving a polynomial (in k) number of linear programs will be a $O(\log k)$ approximation for the Max-Min Congestion Problem. ■

5.3. Constructions for Vertex Sparsifiers

Lemma 7: There is a polynomial time algorithm to find a Type 2 Inequality that is within an $O(\log^2 k)$ factor approximately the maximally violated Type 2 Inequality.

Proof: Let A be the maximally violated Type 2 Inequality. Let $OPT = \frac{h_K(A)}{h'(A)}$. Then using Lemma 4 there is a maximum concurrent flow demand vector $\vec{d}$ such that $\vec{d}$ can be routed with congestion at most 1 in G and cannot be routed with congestion $< OPT$ in G'. So the solution to the Max-Min Congestion Problem is at least OPT, which implies that we will find a demand vector $\vec{d}$ which can be routed in G with congestion at most 1 and cannot be routed in G' with congestion $< \frac{OPT}{O(\log k)}$.

Hence we can find a cut $A \subset K$ for which

$$\frac{h'(A)}{d(A)} < \frac{O(\log^2 k)}{OPT} \text{ and } \frac{h_K(A)}{d(A)} \geq 1$$

and this implies

$$\frac{h_K(A)}{h'(A)} > \frac{OPT}{\log^2 k}$$

■

Returning to the linear program for finding such a G', we can use the ellipsoid algorithm to find (in time polynomial in n and k) a point $\vec{x} \geq 0$ such that the graph G' defined by these edge weights satisfies:

$$\frac{h_K(A)}{g(k)} \leq \sum_{a,bs.t.a\in A,b\notin A} x_{a,b} \leq f(k)h_K(A)$$

for $g(k) = O(\log^2 k)$ and $f(k) = \log^{1.5} k$.

Theorem 9: There exist a polynomial (in n and k) time algorithm to construct an undirected, capacitated graph $G' = (K, E')$ such that the cut function h' of this graph $O(\log^{3.5} k)$-approximates the terminal cut function h_K.

6. Applications

6.1. An Improved Approximation for Requirement Cut

Given an undirected, capacitated graph $G = (V, E)$ and g groups of nodes $X_1, ...X_g \subset V$, each group X_i is assigned a requirement $r_i \in \{0, ...|X_i|\}$. Then the goal of the requirement cut problem is to find a minimum capacity set of edges whose removal separates each group X_i into at least r_i disconnected components.

[17] gives a $O(\log n \log gR)$ approximation algorithm for this problem, where R is the maximum requirement $\max_i r_i$. Then given an instance of the requirement cut problem in which $X_1 \cup X_2 ... \cup X_g = K$ and $|K| = k$, we can use Theorem 9 to reduce to a uniform case. Let OPT be the value of the optimal solution in G. We denote the optimal solution in G' as OPT'.

Claim 3:

$$OPT' \leq O(\log^{3.5} k)OPT$$

Proof: Interpret the optimal solution to the requirement cut problem in G as a partition $P = \{P_1, P_2, ...P_r\}$ of K that satisfies the requirement cut - i.e. for all i, the nodes in X_i are contained in at least r_i elements of the partition P. Then

$$OPT \geq \frac{1}{2}\sum_i h_K(P_i)$$

and $P = \{P_1, P_2, ...P_r\}$ is a valid partition for the requirement cut problem mapped to G' so

$$OPT' \leq \sum_i h'(P_i) \leq O(\log^{3.5} k)\sum_i h_K(P_i)$$

■

Note that Steiner generalization of sparsest cut, min-bisection, ρ-separator, also satisfy this type of reducibility property.

Theorem 10: There is a polynomial (in n and k) time $O(\log^{4.5} k \log gR)$-approximation algorithm for the requirement cut problem

Proof: Construct G' as in Theorem 9 and we can run the approximation algorithm due to [17] to find a set of edges of capacity at most $C \leq O(\log k \log gR)OPT'$ deleting which (in G') results in a partition $P' = \{P'_1, P'_2, ...P'_q\}$ that satisfies the requirement cut. Then

$$\sum_i h'(P'_i) = 2C$$

For each i, define F_i as a set of edges in G that achieves $h_K(P'_i)$ and separates P'_i and $K - P'_i$. Delete all edges in $F = F_1 \cup F_2 ... \cup F_q$, and this results (in G) in a sub-partition P'' of P' that also satisfies the requirement cut and the capacity of these edges is at most $2C$. ■

An almost identical argument that uses the result due to [10] implies:

Corollary 1: There is a polynomial (in n and k) time $O(\log^{4.5} k)$-approximation algorithm for the l-multicut problem

Note that here k is the number of demand pairs. Previous approximation algorithms for these problems [17], [10] and later [19] all rely on a decomposition tree for the graph G that approximates the cuts and such a decomposition tree cannot approximate cuts better than the $\Omega(\log n)$ lower bound for oblivious routing. But we were able to use a black-box reduction to the uniform case to get an approximation guarantee that is $poly(\log k)$.

6.2. Applications to Linear Arrangements

We give another application to illustrate the uses of Theorem 9. The minimum cut linear arrangement problem is defined as: Given an undirected, capacitated graph $G = (V, E)$ we want to find an ordering of the vertices $v_1, v_2, ...v_n$ which minimizes the value of

$$C = \max_{1 \le i \le n} h(\{v_1, v_2, ...v_i\})$$

We can define a natural generalization of this problem in which we are given a set $K \subset V$ of size k, and we want to find an ordering of the nodes in K, $u_1, u_2, ...u_k$ and a partition $A_1, A_2, ...A_k$ of the remaining nodes $V - K$ (and let $B_i = A_i \cup \{u_i\}$) which minimizes the value of

$$C_K = \max_{1 \le i \le k} h(\cup_{1 \le j \le i} B_i)$$

We refer to this problem as the *Steiner Min-Cut Linear Arrangement Problem*. We can also give an identical generalization of minimum linear arrangement to the *Steiner Minimum Linear Arrangement Problem*.

Applying our generic reduction procedure to these problems will require a more intricate uncrossing argument (that relies on the sub-modularity of the cut function) to map a solution in G' back to a solution in G. But the reduction procedure is quite robust, and will work in this setting too: Suppose that the optimal solution to the Steiner Min-Cut Linear Arrangement Problem has value OPT. Again we construct a graph G' as in Theorem 9. Let OPT' be the value of an optimal solution to the min-cut linear arrangement problem on G'.

Claim 4:

$$OPT' \le O(\log^{3.5} k) OPT$$

Proof: Suppose that the optimal solution to the Steiner Min-Cut Linear Arrangement Problem in G has an ordering $u_1, u_2, ...u_k$ of the nodes in K. Then consider this ordering as a solution to the min-cut linear arrangement problem in G'.

Each set $\cup_{1 \le j \le i} B_i$ defines a cut in G that separates $\{u_1, u_2, ...u_i\}$ from $\{u_{i+1}, u_{i+2}, ...u_k\}$. So $h_K(\{u_1, u_2, ...u_i\}) \le h(\cup_{1 \le j \le i} B_i)$ and so $h'(\{u_1, u_2, ...u_i\}) \le O(\log^{3.5} k) h(\cup_{1 \le j \le i} B_i)$ and this is true for all $\{u_1, u_2, ...u_i\}$.

Because for all $\{u_1, u_2, ...u_i\}$, $h(\cup_{1 \le j \le i} B_i) \le OPT$ this implies that for all $\{u_1, u_2, ...u_i\}$

$$h'(\{u_1, u_2, ...u_i\}) \le O(\log^{3.5} k) OPT$$

$$OPT' \le \max_{1 \le i \le k} h'(\{u_1, u_2, ...u_i\}) \le O(\log^{3.5} k) OPT$$

■

Suppose we can find a min-cut linear arrangement of G' of value C'.

Claim 5: We can find a solution to the Steiner Min-Cut Linear Arrangement Problem in G of value at most C'.

Proof: The cut function $h : 2^V \to \Re^+$ is a submodular function. So for all $S, T \subset V$:

$$h(S) + h(T) \ge h(S \cap T) + h(S \cup T)$$

So consider a solution to the min-cut linear arrangement problem in G' of value C'. And suppose that the ordering for K is $\{u_1, u_2, ...u_k\}$. For each i, find a set $B_i \subset V$ s.t. $h(B_i) = h_K(\{u_1, u_2, ...u_i\})$.

Consider the sets B_1 and B_2. We can find sets B_1', B_2' such that $B_1' \subset B_2'$ and $h(B_1) = h(B_1'), h(B_2) = h(B_2')$ and $B_1' \cap K = B_1 \cap K$, $B_2' \cap K = B_2 \cap K$. This is true via submodularity: Choose $B_1' = B_1 \cap B_2$ and $B_2' = B_1 \cup B_2$.

So $B_1' \cap K = (B_1 \cap K) \cap (B_2 \cap K) = (B_1 \cap K)$ and also $B_2' \cap K = (B_1 \cap K) \cup (B_2 \cap K) = B_2 \cap K$ because $B_1 \cap K \subset B_2 \cap K$.

Also

$$h(B_1) + h(B_2) \ge h(B_1') + h(B_2')$$

via submodularity. However $h(B_1)$ is the minimal value of an edge cut separating $B_1 \cap K$ from $K - (B_1 \cap K)$ and B_1' also separates $B_1 \cap K$ from $K - (B_1 \cap K)$, and a similar statement holds for B_2'. So the above inequality implies

$$h(B_1) = h(B_1'), h(B_2) = h(B_2')$$

We can continue the above argument and get sets $B_1', B_2', ...B_k'$ such that

$$h(B_i') = h_K(\{u_1, u_2, ...u_i\}) \text{ and } B_1' \subset B_2' ... \subset B_k'$$

Then we can choose $A_i' = (B_i' - B_{i-1}') \cap (V - K)$ as our partition of $V - K$ (and let $D_i = A_i' \cup \{u_i\}$) and then for any $1 \le i \le k$:

$$h(\cup_{1 \le j \le i} D_i) = h_K(\{u_1, u_2, ...u_i\}) \le h'(\{u_1, u_2, ...u_i\})$$

So

$$\max_{1 \le i \le k} h(\cup_{1 \le j \le i} D_i) \le \max_{1 \le i \le k} h'(\{u_1, u_2, ...u_i\}) \le C'$$

■

Using approximation algorithms due to [15] for min-cut linear arrangement and due to [6] for minimum linear arrangement, we get:

Theorem 11: There is a polynomial (in n and k) time $O(\log^4 k \log\log k)$-approximation algorithm for the Steiner Minimum Linear Arrangement Problem, and a polynomial time $O(\log^4 k \log\log k)$-approximation algorithm for the Steiner Minimum Linear Arrangement Problem.

6.3. Applications to 0*-Extensions*

Theorem 12: There is a polynomial time constructible $O(\log^{4.5} k)$-competitive oblivious algorithm for the 0-extension problem.

We defer the proof, and additional applications to the full version.

7. Improvements and an Impossibility Theorem

In joint work with Tom Leighton, we have made improvements to the structural and algorithmic results in this paper. These results mostly use the ideas introduced here, and improvements come from considering a duality inspired zero-sum game and from adapting techniques from [4] in the construction of such a G'.

For any multicommodity demand $\vec{f} \in \Re^{\binom{k}{2}}$, we denote $cong_G(\vec{f})$ as the minimum congestion needed to route $\vec{f}$ (interpreted as demands on pairs of nodes in K) in G. We prove there exists a graph $G' = (K, E')$ such that

$$cong_{G'}(\vec{f}) \leq cong_G(\vec{f}) \leq O(\frac{\log k}{\log\log k}) cong_{G'}(\vec{f})$$

for all demands $\vec{f} \in \Re^{\binom{k}{2}}$. This structural result implies the structural result given in this paper. We also give a polynomial (in n and k) time algorithm to construct a graph $G' = (K, E')$ for which

$$cong_{G'}(\vec{f}) \leq cong_G(\vec{f}) \leq O(\frac{\log^2 k}{\log\log k}) cong_{G'}(\vec{f})$$

for all demands $\vec{f} \in \Re^{\binom{k}{2}}$.

We also prove an $\Omega(\log\log k)$ lower bound: We give a graph $G = (V, E)$ and a subset $K \subset V$ of size k for which any graph $G' = (K, E')$ which satisfies (for all $\vec{f} \in \Re^{\binom{k}{2}}$) $cong_{G'}(\vec{f}) \leq cong_G(\vec{f})$, there must be a demand $\vec{f} \in \Re^{\binom{k}{2}}$ for which $cong_G(\vec{f}) > \Omega(\log\log k) cong_{G'}(\vec{f})$. This gives a lower bound on the cost of vertex sparsifiers for multicommodity demands. No such lower bounds were previously known, despite the known lower bounds on the integrality gap for the linear programming relaxation of the 0-extension problem.

Acknowledgments

We would like to thank Yossi Azar, Tom Leighton, Harald Räcke, Satish Rao and Mohit Singh for many helpful discussions.

References

[1] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor, "A general approach to online network optimization problems," *Symposium on Discrete Algorithms*, pp. 577–586, 2004.

[2] S. Arora, J. Lee, and A. Naor, "Euclidean distortion and the sparsest cut," *Symposium on Theory of Computing*, pp. 553–562, 2005.

[3] Y. Aumann and Y. Rabani, "An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm," *SIAM Journal on Computing*, vol. 27, pp. 291–301, 1998.

[4] Y. Azar, E. Cohen, A. Fiat, K. Haim, and H. Räcke, "Optimal oblivious routing in polynomial time," *Symposium on Theory of Computing*, pp. 383–388, 2003.

[5] G. Calinescu, H. Karloff, and Y. Rabani, "Approximation algorithms for the 0-extension problem," *Symposium on Discrete Algorithms*, pp. 8–16, 2001.

[6] M. Charikar, M. Hajiaghayi, H. Karloff, and S. Rao, "ℓ_2^2 spreading metrics for vertex ordering problems," *Symposium on Discrete Algorithms*, pp. 1018–1027, 2006.

[7] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar, "An improved approximation algorithm for the 0-extension problem," *Symposium on Discrete Algorithms*, pp. 257–265, 2003.

[8] J. Fakcharoenphol, S. Rao, and K. Talwar, "A tight bound on approximating arbitrary metrics by tree metrics," *Symposium on Theory of Computing*, pp. 448–455, 2003.

[9] N. Garg, V. Vazirani, and M. Yannakakis, "Approximate max-flow min-(multi)cut theorems and their applications," *SIAM Journal on Computing*, vol. 25, pp. 235–251, 1996.

[10] D. Golovin, V. Nagarajan, and M. Singh, "Approximating the k-multicut problem," *Symposium on Discrete Algorithms*, pp. 621–630, 2006.

[11] M. Grötschel, L. Lovász, and A. Schrijver, "Geometric algorithms and combinatorial optimization," *Springer Verlag*, 1993.

[12] C. Harrelson, K. Hildrum, and S. Rao, "A polynomial-time tree decomposition to minimize congestion," *Symposium on Parallel Algorithms*, pp. 34–43, 2003.

[13] A. Karzanov, "Minimum 0-extensions of graph metrics," *European Journal of Combinatorics*, pp. 71–101, 1998.

[14] P. Klein, S. Plotkin, and S. Rao, "Excluded minors, network decomposition, and multicommodity flow," *Symposium on Theory of Computing*, pp. 682–690, 1993.

[15] L. C. Lau, "An approximate max-steiner-tree packing min-steiner-cut theorem," *Combinatorica*, pp. 71–90, 2007.

[16] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *Journal of the ACM*, pp. 787–832, 1999.

[17] N. Linial, E. London, and Y. Rabinovich, "The geometry of graphs and some of its algorithmic applications," *Combinatorica*, vol. 15, pp. 215–245, 1995.

[18] V. Nagarajan and R. Ravi, "Approximation algorithms for requirement cut on graphs," *International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pp. 209–220, 2005.

[19] H. Räcke, "Minimizing congestion in general networks," *Foundations of Computer Science*, pp. 43–52, 2003.

[20] H. Räcke, "Optimal hierarchical decompositions for congestion minimization in networks," *Symposium on Theory of Computing*, pp. 255–264, 2008.