Vertex Sparsifiers and Abstract Rounding Algorithms

Moses Charikar*
Princeton University
moses@cs.princeton.edu

Tom Leighton
MIT and Akamai Technologies, Inc
ftl@math.mit.edu

Shi Li*
Princeton University
shili@cs.princeton.edu

Ankur Moitra[†]
MIT
moitra@mit.edu

Abstract—The notion of vertex sparsification (in particular cut-sparsification) is introduced in [18], where it was shown that for any graph $G = (V, E)$ and any subset of k terminals $K \subset V$, there is a polynomial time algorithm to construct a graph $H = (K, E_H)$ *on just the terminal set* so that simultaneously for all cuts $(A, K - A)$, the value of the minimum cut in G separating A from $K - A$ is approximately the same as the value of the corresponding cut in H. Then approximation algorithms can be run directly on H as a proxy for running on G.

We give the first super-constant lower bounds for how well a cut-sparsifier H can simultaneously approximate all minimum cuts in G. We prove a lower bound of $\Omega(\log^{1/4} k)$ – this is polynomially-related to the known upper bound of $O(\log k / \log \log k)$. Independently, a similar lower bound is given in [17]. This is an exponential improvement on the $\Omega(\log \log k)$ bound given in [14] which in fact was for a stronger vertex sparsification guarantee, and did not apply to cut sparsifiers.

Despite this negative result, we show that for many natural optimization problems, we do not need to incur a multiplicative penalty for our reduction. Roughly, we show that any rounding algorithm which also works for the 0-extension relaxation can be used to construct good vertex-sparsifiers for which the optimization problem is easy. Using this, we obtain optimal $O(\log k)$-competitive Steiner oblivious routing schemes, which generalize the results in [20]. We also demonstrate that for a wide range of graph packing problems (which includes maximum concurrent flow, maximum multiflow and multicast routing, among others, as a special case), the integrality gap of the linear program is always at most $O(\log k)$ times the integrality gap restricted to trees. Lastly, we use our ideas to give an efficient construction for vertex-sparsifiers that match the current best existential results – this was previously open. Our algorithm makes novel use of Earth-mover constraints.

Keywords-vertex sparsifier; approximation algorithms;

I. Introduction

A. Background

The notion of vertex sparsification (in particular cut-sparsification) is introduced in [18]: Given a graph $G = (V, E)$ and any subset of terminals $K \subset V$, the goal is to construct a graph $H = (K, E_H)$ *on just the terminal set* so that simultaneously for all cuts $(A, K - A)$, the value of the minimum cut in G separating A from $K - A$ is approximately the same as the value of the corresponding cut in H. If for all cuts

$(A, K - A)$, the value of the cut in H is at least the value of the corresponding minimum cut in G and is at most α times this value, then we call H a cut-sparsifier of quality α. Throughout this paper we will let $|V| = n$ and $|K| = k$, and we think of $k << n$.

The motivation for considering such questions is in obtaining approximation algorithms with guarantees that are independent of the size of the graph. For many graph partitioning and multicommodity flow questions, the value of the optimum solution is determined (or can be approximated) given just the values of the minimum cut separating A from $K - A$ in G (for *every* $A \subset K$). As a result the value of the optimum solution is approximately preserved when mapping the optimization problem to H. So approximation algorithms can be run on H as a proxy for running directly on G, and because the size (number of nodes) of H is k, any approximation algorithm that achieves a $poly(\log n)$-approximation guarantee in general will achieve a $poly(\log k)$ approximation guarantee when run on H (provided that the quality α is also $poly(\log k)$). Feasible solutions in H can also be mapped back to feasible solutions in G for many of these problems, so polynomial time constructions for good cut-sparsifiers yield black box techniques for designing approximation algorithms with guarantees $poly(\log k)$ (and independent of the size of the graph).

In addition to being useful for designing approximation algorithms with improved guarantees, the notion of cut-sparsification is also a natural generalization of many methods in combinatorial optimization that attempt to preserve certain cuts in G (as opposed to all minimum cuts) on a simpler graph H - for example Gomory-Hu Trees, and Mader's Theorem. Here we consider a number of questions related to cut-sparsification:

1) Is there a super-constant lower bound on the quality of cut-sparsifiers? Do the best (or even near-best) cut-sparsifiers necessarily result from a distribution on contractions?

2) Do we really need to pay a price in the approximation guarantee when applying vertex sparsification techniques to an optimization problem?

3) Can we efficiently construct cut-sparsifiers with quality as good as the current best *existential* results?

We resolve all of these questions in this paper. In the

preceding subsections, we will describe what is currently known about each of these questions, our results, and our techniques.

B. Super-Constant Lower Bounds and Separations

In [18], it is proven that in general there are always cut-sparsifiers H of quality $O(\log k / \log \log k)$. In fact, if G excludes any fixed minor then this bound improves to $O(1)$. Yet prior to this work, no super-constant lower bound was known for the quality of cut-sparsifiers in general. We prove

Theorem 1. *There is an infinite family of graphs that admits no cut-sparsifiers of quality better than $\Omega(\log^{1/4} k)$.*

Independent of our work, Makarychev and Makarychev obtained an $\tilde{\Omega}(\log^{1/4} k)$ lower bound for cut-sparsifiers, and an $\tilde{\Omega}(\log^{1/2} k)$ lower bound for flow-sparsifiers, and this nearly matches the current best integrality gap of the 0-extension linear programming relaxation [12], [5]. [17] also gives an exciting connection between vertex sparsification and the lipschitz extendability of Banach spaces, which has been an actively studied in functional analysis since the 1950s.

Some results are known in more general settings. In particular, one could require that the graph H not only approximately preserve minimum cuts but also approximately preserve the congestion of all multicommodity flows (with demands endpoints restricted to be in the terminal set). This notion of vertex-sparsification is referred to as flow-sparsification (see [14]) and admits a similar definition of quality. [14] gives a lower bound of $\Omega(\log \log k)$ for the quality of flow-sparsifiers. However, this does not apply to cut sparsifiers and in fact, for the example given in [14], there is an $O(1)$-quality cut-sparsifier!

Our bound is polynomially related to the current best existential upper-bound [18], which is $O(\log k / \log \log k)$. We note that in the current best existential upper bounds, good vertex sparsifiers are actually generated as a convex combination of contractions on the base graph G. As it turns out, lower bounds against restricted vertex sparsifiers (i.e. generated from a convex combination of contractions) follow immediately from integrality gaps for a natural LP relaxation of the 0-extension problem. Yet what takes the most work in proving quality lower bounds is removing an assumption on how the best quality vertex sparsifier can be generated.

In this paper, we also give the first super-constant separation between contraction based vertex sparsification, and unrestricted vertex sparsification:

Theorem 2. *There is an infinite family of graphs so that the quality of the best cut-sparsifier is asymptotically better than the quality of the best cut-sparsifer that can be generated through contractions.*

The bound we obtain in the above theorem is polynomial

in $\log \log \log k$. Our approach may be of independent interest: we use Bourgain's Junta Theorem [4], and the Hypercontractive Inequality [3], [2] to analyze the quality of a particular cut-sparsifier for a hypercube-like base graph G.

C. Abstract Integrality Gaps and Rounding Algorithms

As described earlier, running an approximation algorithm on the sparsifier $H = (K, E_H)$ as a proxy for the graph $G = (V, E)$ pays an additional price in the approximation guarantee that corresponds to how well H approximates G. Here we consider the question of whether this loss can be avoided.

As a motivating example, consider the problem of Steiner oblivious routing [18]. Previous techniques for constructing Steiner oblivious routing schemes [18], [14] first construct a vertex sparsifier H for G, construct an oblivious routing scheme in H and then map this back to a Steiner oblivious routing scheme in G. Any such approach must pay a price in the competitive ratio, and cannot achieve an $O(\log k)$-competitive guarantee because (for example) expanders do not admit constant factor flow-sparsifiers [14].

So black box reductions pay a price in the competitive ratio, yet here we present a technique for *combining* the flow-sparsification techniques in [14] and the oblivious routing constructions in [20] into a single step, and we prove that there are $O(\log k)$-competitive Steiner oblivious routing schemes, which is optimal. This result is a corollary of a more general idea:

The constructions of flow-sparsifiers given in [14] (which is an extension of the techniques in [18]) can be regarded as a dual to the rounding algorithm in [8] for the 0-extension problem. What we observe here is: Suppose we are given a rounding algorithm that is used to round the fractional solution of some relaxation to an integral solution for some optimization problem. If this rounding algorithm also works for the relaxation for the 0-extension problem given in [12] (and also used in [5], [8]), then we can use the techniques in [18], [14] to obtain *stronger* flow-sparsifiers which are not only good quality flow-sparsifiers, but also for which the optimization problem is easy. So in this way we do not need to pay an additional price in the approximation guarantee in order to replace the dependence on n with a dependence on k. With these ideas in mind, what we observe is that the rounding algorithm in [9] wh ich embeds metric spaces into distributions on dominating tree-metrics, can also be used to round the 0-extension relaxation. This allows us to construct flow-sparsifiers that have $O(\log k)$ quality, and at the same time can be explicitly written as a convex combination of 0-extensions that are tree-like. On trees, oblivious routing is easy, and so this gives us a way to simultaneously construct good flow-sparsifiers and good oblivious routing schemes on the sparsifier in one step! This upper bound was obtained independently in [7], who also

observe that one can ask for vertex sparsifiers to be simple, as well as good approximations to the original graph.

Of course, the rounding algorithm in [9] for embedding metric spaces into distributions on dominating tree-metrics is a *very* common first step in rounding fractional relaxations of graph partitioning, graph layout and clustering problems. So for all problems that use this embedding as the main step, we are able to replace the dependence on n with dependence on k, and we do not introduce any additional poly-logarithmic factors as in previous work! One can also interpret our result as giving a generalization of the hierarchical decompositions given in [20] for approximating the cuts in a graph G on trees. We state our results more formally, below, and we refer to such a statement as an ABSTRACT INTEGRALITY GAP.

Definition 1. *We call a fractional packing problem P a graph packing problem if the goal of the dual (covering) problem D is to minimize the ratio of the total units of distance $\times$ capacity allocated in the graph divided by some monotone increasing function of the distances between terminals.*

This definition is quite general, and captures maximum concurrent flow, maximum multiflow, and multicast routing as special cases, in addition to many other common optimization problems. The integral[1] dual *ID* problems are generalized sparsest cut, multicut and requirement cut respectively.

Theorem 3. *For any graph packing problem P, the maximum ratio of the integral dual to the fractional primal is at most $O(\log k)$ times the maximum ratio restricted to trees.*

For a packing problem that fits into this class, this theorem allows us to reduce bounding the integrality gap in general graphs to bounding the integrality gap on trees, which is often substantially easier than for general graphs (i.e. for the example problems given above). We believe that this result helps to explain the ubiquity of the $O(\log k)$ bound for the flow-cut gap for a wide range of multicommodity flow problems.

We also give a polynomial time algorithm to reduce any graph packing problem P to a corresponding problem on a tree: Again, let K be the set of terminals.

Definition 2. *Let $OPT(P,G)$ be the optimal value of the fractional graph packing problem P on the graph G.*

Theorem 4. *There is a polynomial time algorithm to construct a distribution μ on (a polynomial number of) trees on the terminal set K, s.t. $E_{T\leftarrow\mu}[OPT(P,T)] \leq O(\log k)OPT(P,G)$ and such that any valid integral dual*

of cost C (for any tree T in the support of μ) can be immediately transformed into a valid integral dual in G of cost at most C.

As a corollary, given an approximation algorithm that achieves an approximation ratio of C for the integral dual to a graph packing problem on trees, we obtain an approximation algorithm with a guarantee of $O(C\log k)$ for general graphs. We will refer to this last result as an ABSTRACT ROUNDING ALGORITHM.

D. Improved construction of flow-sparsifiers

We also give a polynomial time construction of $O(\log k/\log\log k)$ quality flow-sparsifiers (and consequently cut-sparsifiers as well), which were previously only known to exist, but finding a polynomial time construction was still open. We accomplish this by performing a lifting (inspired by Earth-mover constraints) on an appropriate linear program. This lifting allows us to implicitly enforce a constraint that previously was difficult to enforce, and required an approximate separation oracle rather than an exact separation oracle. Alternative constructions that achieve this bound were also independently given in [17] and [7]. We give the details in Section V.

II. PRELIMINARIES

A. Maximum concurrent flow

An instance of the maximum concurrent flow problem consists of an undirected graph $G = (V,E)$, a capacity function $c : E \rightarrow \Re^+$ that assigns a non-negative capacity to each edge, and a set of demands $\{(s_i, t_i, f_i)\}$ where $s_i, t_i \in V$ and f_i is a non-negative demand. We denote $K = \cup_i\{s_i, t_i\}$. 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 and the corresponding dual can be written as :

$$
\begin{array}{ll}
\max \quad \lambda & \min \quad \sum_e d(e)c(e) \\
\text{s.t.} & \text{s.t.} \\
\sum_{P\in P_{s_i,t_i}} x(P) \geq \lambda f_i & \forall_{P\in P_{s_i,t_i}} \sum_{e\in P} d(e) \geq D(s_i,t_i) \\
\sum_{P\ni e} x(P) \leq c(e) & \sum_i D(s_i,t_i)f_i \geq 1 \\
x(P) \geq 0 & d(e) \geq 0, D(s_i,t_i) \geq 0
\end{array}
$$

For a maximum concurrent flow problem, let λ^* denote the optimum.

Let $|K| = k$. Then for a given set of demands $\{s_i, t_i, f_i\}$, we associate a vector $\vec{f} \in \Re^{\binom{k}{2}}$ in which each coordinate corresponds to a pair $(x,y) \in \binom{K}{2}$ and the value $\vec{f}_{x,y}$ is defined

as the demand f_i for the terminal pair $s_i = x, t_i = y$. Also given a capacitated graph $H = (K, E_H)$ we will write $\vec{H} \in \mathfrak{R}^{\binom{k}{2}}$ for the demand vector in which each coordinate (which corresponds to a pair $a, b \in K$) is set to the capacity of the edge $(a, b) \in E_H$.

Definition 3. *We denote* $cong_G(\vec{f}) = \frac{1}{\lambda^*}$

Or equivalently $cong_G(\vec{f})$ is the minimum C s.t. $\vec{f}$ can be routed in G and the total flow on any edge is at most C times the capacity of the edge.

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).

B. Cut Sparsifiers

Suppose we are given an undirected, capacitated graph $G = (V, E)$ and a set $K \subset V$ of terminals of size k. Let $h : 2^V \to \mathfrak{R}^+$ 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 $h_K : 2^K \to \mathfrak{R}^+$ which we refer to as the terminal cut function on K: $h_K(U) = \min_{A \subset V \text{ s.t. } A \cap K = U} h(A)$.

Definition 4. *G' is a* cut-sparsifier *for the graph $G = (V, E)$ and the terminal set K if G' is a graph on just the terminal set K (i.e. $G' = (K, E')$) and if the cut function $h' : 2^K \to \mathfrak{R}^+$ of G' satisfies (for all $U \subset K$) $h_K(U) \leq h'(U)$.*

We can define a notion of quality for any particular cut-sparsifier:

Definition 5. *The* quality *of a cut-sparsifier G' is defined as $max_{U \subset K} h'(U)/h_K(U)$.*

C. 0-Extensions

Definition 6. *$f : V \to K$ is a* 0-extension *if for all $a \in K$, $f(a) = a$.*

So a 0-extension f is a clustering of the nodes in V into sets, with the property that each set contains exactly one terminal.

Definition 7. *Given a graph $G = (V, E)$ and a set $K \subset V$, and 0-extension f, $G_f = (K, E_f)$ is a capacitated graph in which for all $a, b \in K$, the capacity $c_f(a, b)$ of edge $(a, b) \in E_f$ is*

$$\sum_{(u,v) \in E \text{ s.t. } f(u) = a, f(v) = b} c(u, v)$$

D. Steiner Oblivious Routing

Given a graph $G = (V, E)$ with capacity function c, and a set of terminals $K \subset V$, a Steiner oblivious routing scheme is a set of routings $\{R_{s,t} : s, t \in K\}$, where each routing $R_{s,t}$ sends 1 unit flow from s to t. We say $\{R_{s,t}\}$ has competitive ratio α if for every set of demands $\{(s_i, t_i, f_i)\}$ where $s_i, t_i \in K$ that can be routed in G (with congestion 1), the routing $\sum_i f_i R_{s_i, t_i}$ has congestion at most α. This generalize the defintion of oblivious routing schemes, in which case $K = V$.

III. LOWER BOUNDS FOR CUT SPARSIFIERS

Consider the following construction for a graph G. Let Y be the hypercube of size 2^d for $d = \log k$. Then for every node $y_s \in Y$ (i.e. $s \in \{0, 1\}^d$), we add a terminal z_s and connect the terminal z_s to y_s using an edge of capacity $\sqrt{d}$. All the edges in the hypercube are given capacity 1. We will use this instance to show two lower bounds, one for cut-sparsifiers generated from contractions, and one for unrestricted cut-sparsifiers.

A. Lower bound for Cut Sparsifiers from 0-extensions

In this subsection, we give a particular relaxation for the 0-extension problem (when the input metric is ℓ_1). We show a strong duality relation between the worst case integrality gap for this relaxation, and the quality of the best cut-sparsifier that can result from a distribution on contractions. So lower bounds for cut-sparsifiers are as easy as integrality gaps.

We also give an $\Omega(\sqrt{d})$ integrality gap for this relaxation, on the example above. A similar bound is actually implicit in the work of [11] too. This integrality gap immediately implies a lower bound for contraction-based cut-sparsifiers for the above graph.

Given the graph $G = (V, E)$ a set $K \subset V$ of terminals, and a semi-metric D on K the 0-extension problem [12] is defined as:

Definition 8. *The* **0-Extension Problem** *is defined as*

$$\min_{\text{0-Extensions} f} \sum_{(u,v) \in E} c(u, v) D(f(a), f(b))$$

We denote $OPT(G, K, D)$ as the value of this optimum.

Definition 9. *Let Δ_U denote the cut-metric in which $\Delta_U(u, v) = 1_{|U \cap \{u,v\}| = 1}$.*

If we are given a semi-metric D which is ℓ_1, we can define an (exponential) sized linear program that will be useful for analyzing cut-sparsifiers that arise from contractions:

$$\min \quad \sum_U \delta(U) h(U)$$
$$\text{s.t.}$$
$$\forall_{t,t' \in K} \sum_U \delta(U) \Delta_U(t, t') = D(t, t').$$

We will refer to this linear program as the **Cut-Cut Relaxation**. For a particular instance (G, K, D) of the 0-extension problem, we denote the optimal solution to this linear program as $OPT_{cc}(G, K, D)$.

Definition 10. *The* Contraction Quality *of G, K is the minimum α such that there is a distribution on 0-extensions γ and $H = \sum_f \gamma(f) G_f$ is a α quality cut-sparsifier.*

Lemma 1. *Let ν be the maximum integrality gap of the Cut-Cut Relaxation for a particular graph $G = (V, E)$, a particular set $K \subset V$ of terminals, over all ℓ_1 semi-metrics D on K. Then the Contraction Quality of G, K is exactly ν.*

Let α be the Contraction Quality of G, K. Then [18] demonstrates that $\alpha \leq \nu$. In the other direction: Suppose γ is a distribution on 0-extensions s.t. $H = \sum_f \gamma(f) G_f$ is a α-quality cut sparsifier. In a sense, because H is a good cut-sparsifier γ is a good *oblivious* algorithm for the 0-extension problem restricted to cut-metrics. This in turn implies that γ is a good *oblivious* algorithm for convex combinations of cut-metrics (i.e. ℓ_1 metrics) and this will imply a bound on the integrality gap of the Cut-Cut Relaxation for the 0-extension problem on G, K when the input metric is ℓ_1. We defer the details to the full version of our paper.

Consider the following distance assignment to the edges in our example graph given at the beginning of this section: Each edge connecting a terminal to a node in the hypercube - i.e. an edge of the form (z_s, y_s) is assigned distance $\sqrt{d}$ and every other edge in the graph is assigned distance 1. Then let σ be the shortest path metric on V given these edge distances.

Claim 1. *σ is an ℓ_1 semi-metric on V, and in fact there is a weighted combination of cuts s.t. $\sigma(u,v) = \sum_U \delta(U) \Delta_U(u,v)$ and $\sum_U \delta(U) h(U) = O(kd)$*

Yet if we take D equal to the restriction of σ on K, then $OPT(G, K, D) = \Omega(kd^{3/2})$:

Lemma 2. *$OPT(G, K, D) = \Omega(kd^{3/2})$*

The proof of this lemma is based on the small-set expansion properties of the hypercube, and we defer a proof to the full version of our paper.

B. Lower bounds for Arbitrary Cut sparsifiers

We will in fact use the above example graph G to give a lower bound on the quality of *any* cut-sparisifer. We will show that for this graph, no cut-sparsifier achieves quality better than $\Omega(\log^{1/4} k)$.

The particular example G that we gave above has many symmetries - i.e. there are many automorphisms of K that exactly preserve the terminal cut function h_K. We can use these automorphisms to symmetrize good cut-sparsifiers without degrading the quality, and as a result we can assume that the best cut-sparsifier is highly symmetric:

Claim 2. *If α is the best quality cut-sparsifier for the above graph G, then there is an α quality cut-sparsifier H in which the capacity between two terminals z_s and z_t is only dependent on the Hamming distance $Hamm(s,t)$.*

One can regard any cut-sparsifier (not just ones that result from contractions) as a set of $\binom{k}{2}$ variables, one for the capacity of each edge in H. Then the constraints that H be an α-quality cut-sparsifier are just a system of inequalities, one for each subset $A \subset K$ that enforces that the cut in H is at least as large as the minimum cut in G (i.e. $h'(A) \geq h_K(A)$) and one enforcing that the cut is not too large (i.e. $h'(A) \leq \alpha h_K(A)$). Then in general, one can derive lower

bounds on the quality of cut-sparsifiers by showing that if α is not large enough, then this system of inequalities is infeasible meaning that there is not cut-sparsifier achieving quality α. Unlike the above argument, this does not assume anything about how the cut-sparsifier is generated.

Theorem 1. *For $\alpha = \Omega(\log^{1/4} k)$, there is no cut-sparsifier H for G which has quality at most α.*

Proof (sketch): Assume that there is a cut-sparsifier H' of quality at most α. Then using the above claim, there is a cut-sparsifier H of quality at most α in which the weight from a to b is only a function of $Hamm(a,b)$. Then for each $i \in [d]$, we can define a variable w_i as the total weight of edges incident to any terminal of length i. I.e. $w_i = \sum_b$ s.t. $Hamm(a,b) = i \, c_H(a,b)$.

For simplicity, here we will assume that all cuts in the sparsifier H are at most the cost of the corresponding minimum cut in G and at least $\frac{1}{\alpha}$ times the corresponding minimum cut. This of course is an identical set of constraints that we get from dividing the standard definition that we use in this paper for α-quality cut-sparsifiers by α.

We need to derive a contradiction from the system of inequalities that characterize the set of α-quality cut sparsifiers for G. As we noted, we will consider only the sub-cube cuts (cuts in which $U = \{z_s \cup y_s | s_1 = s_2 = \cdots s_j = 0\}$) and the Hamming ball $U = \{z_s \cup y_s | d(y_s, y_0) \leq \frac{d}{2}\}$, which we refer to as the Majority Cut.

Consider the Majority Cut: There are $\Theta(k)$ terminals on each side of the cut, and most terminals have Hamming weight close to $\frac{d}{2}$. In fact, we can sort the terminals by Hamming weight and each weight level around Hamming weight $\frac{d}{2}$ has roughly a $\Theta(\frac{1}{\sqrt{d}})$ fraction of the terminals. Any terminal of Hamming weight $\frac{d}{2} - \sqrt{i}$ has roughly a constant fraction of their weight w_i crossing the cut in H, because choosing a random terminal Hamming distance i from any such terminal corresponds to flipping i coordinates at random, and throughout this process there are almost an equal number of 1s and 0s so this process is well-approximated by a random walk starting at $\sqrt{i}$ on the integers, which equally likely moves forwards and backwards at each step for i total steps, and asking the probability that the walk ends at a negative integer.

In particular, for any terminal of Hamming weight $\frac{d}{2} - t$, the fraction of the weight w_i that crosses the Majority Cut is $O(exp\{-\frac{t^2}{i}\})$. So the total weight of length i edges (i.e. edges connecting two terminals at Hamming distance i) cut by the Majority Cut is $O(w_i |\{z_s | Hamm(s, 0) \geq \frac{d}{2} - \sqrt{i}\}|) = O(w_i \sqrt{i/d}) k$ because each weight close to the boundary of the Majority cut contains roughly a $\Theta(\frac{1}{\sqrt{d}})$ fraction of the terminals. So the total weight of edges crossing the Majority Cut in H is $O(k \sum_{i=1}^{d} w_i \sqrt{i/d})$

And the total weight crossing the minimum cut in G separating $A = \{z_s | d(y_s, y_0) \leq \frac{d}{2}\}$ from $K - A$ is $\Theta(k\sqrt{d})$. And

because the cuts in H are at least $\frac{1}{\alpha}$ times the corresponding minimum cut in G, this implies $\sum_{i=1}^{d} w_i \sqrt{i/d} \geq \Omega(\frac{\sqrt{d}}{\alpha})$

Next, we consider the set of sub-cube cuts. For $j \in [d]$, let $A_j = \{z_s | s_1 = 0, s_2 = 0, ..s_j = 0\}$. Then the minimum cut in G separating A_j from $K - A_j$ is $\Theta(|A_j| \min(j, \sqrt{d}))$, because each node in the Hypercube which has the first j coordinates as zero has j edges out of the sub-cube, and when $j > \sqrt{d}$, we would instead choose cutting each terminal $z_s \in A_j$ from the graph directly by cutting the edge (y_s, z_s).

Also, for any terminal in A_j, the fraction of length i edges that cross the cut is approximately $1 - (1 - \frac{j}{d})^i = \Theta(\min(\frac{ij}{d}, 1))$. So the constraints that each cut in H be at most the corresponding minimum cut in G give the inequalities $\sum_{i=1}^{d} \min(\frac{ij}{d}, 1) w_i \leq O(\min(j, \sqrt{d}))$

We refer to the above constraint as B_j. Multiply each B_j constraint by $\frac{1}{j^{3/2}}$ and adding up the constraints yields a linear combination of the variables w_i on the left-hand side. The coefficient of any w_i is $\sum_{j=1}^{d-1} \frac{\min(\frac{ij}{d}, 1)}{j^{3/2}} \geq \sum_{j=1}^{d/i} \frac{\frac{ij}{d}}{j^{3/2}}$.

And using the Integration Rule this is $\Omega(\sqrt{\frac{i}{d}})$.

This implies that the coefficients of the constraint B resulting from adding up $\frac{1}{j^{3/2}}$ times each B_j for each w_i are at least as a constant times the coefficient of w_i in the Majority Cut Inequality. So we get $\sum_{j=1}^{d-1} \frac{1}{j^{3/2}} \min(j, \sqrt{d}) \geq$ $\Omega\left(\sum_{j=1}^{d-1} \frac{1}{j^{3/2}} \sum_{i=1}^{d} \min(\frac{ij}{d}, 1) w_i\right) \geq \Omega\left(\sum_{i=1}^{d} w_i \sqrt{\frac{i}{d}}\right) \geq \Omega\left(\frac{\sqrt{d}}{\alpha}\right)$.

And we can evaluate the constant $\sum_{j=1}^{d-1} j^{-3/2} \min(j, \sqrt{d}) = \sum_{j=1}^{\sqrt{d}} j^{-1/2} + \sqrt{d} \sum_{j=\sqrt{d}+1}^{d-1} j^{-3/2}$ using the Integration Rule, this evaluates to $O(d^{1/4})$. This implies $O(d^{1/4}) \geq \frac{\sqrt{d}}{\alpha}$ and in particular this implies $\alpha \geq \Omega(d^{1/4})$. So the quality of the best cut-sparsifier for H is at least $\Omega(\log^{1/4} k)$. □

This bound is not as good as the lower bound we obtained earlier in the restricted case in which the cut-sparsifier is generated from contractions. As we will demonstrate, there are actually cut-sparsifiers that achieve quality $o(\sqrt{\log k})$ for G.

IV. Noise Sensitive Cut-Sparsifiers

Here we give a cut-sparsifier H that achieves quality $o(\sqrt{\log k})$ for the graph G given in Section III, which is asymptotically better than the quality of the best cut-sparsifier that can be generated from contractions. And so in general restricting to convex combinations of 0-extensions is sub-optimal. This is the first super-constant separation between contraction-based vertex sparsification and unrestricted vertex sparsification.

In G, the minimum cut separating any singleton terminal $\{z_s\}$ from $K - \{z_s\}$ is just the cut that deletes the edge (z_s, y_s). So the capacity of this cut is $\sqrt{d}$. We want a good cut-sparsifier to approximately preserve this cut, so the total capacity incident to any terminal in H will also be $\sqrt{d}$.

We distribute this capacity incident to z_s evenly among all terminals that are Hamming distance (roughly) $\sqrt{d}$ from z_s. This choice of ρ corresponds to flipping each bit in t with probability $\Theta(\frac{1}{\sqrt{d}})$ when generating u from t. We prove that the graph H has cuts at most the corresponding minimum-cut in G. In fact, a stronger statement is true: $\vec{H}$ can be routed as a flow in G with congestion $O(1)$. We prove this fact in the full version of our paper by explicitly constructing a good routing scheme using a "canonical-paths" type argument.

So we know that the cuts in H are never larger than the corresponding minimum cut in G, and all that remains is to show that the cuts in H are never too small. We conjecture that the quality of H is actually $\Theta(\log^{1/4} k)$, and this seems natural since the quality of H restricted to the Majority Cut and the sub-cube cuts is actually $\Theta(\log^{1/4} k)$, and often the Boolean functions corresponding to these cuts serve as extremal examples in the harmonic analysis of Boolean functions. In fact, our lower bound on the quality of any cut-sparsifier for G is based only on analyzing these cuts so in a sense, our lower bound is tight given the choice of cuts in G that we used to derive infeasibility in the system of inequalities characterizing α-quality cut-sparsifiers.

We analyze the quality of H by relating the total capacity of edges in H crossing the cut $(A, K - A)$ to the spectrum of a Boolean function $(f_A : \{-1, +1\}^d \rightarrow \{-1, +1\}$ s.t. $f_A(s) = +1$ iff $z_s \in A)$ associated with the cut. Intuitively, the total capacity of edges crossing the cut $(A, K - A)$ should be related to the noise-sensitivity of the function f_A because given a terminal $u \in K$, choosing a random neighbor of u in H corresponds roughly to flipping $\sqrt{d}$ random coordinates in the binary representation of u to obtain a neighbor v.

We use a case analysis centered around Bourgain's Junta Theorem [4], and the Hypercontractive Inequality [3], [2] to establish that H has quality $o(\sqrt{\log k})$: The minimum cut in G separating A from $K - A$ is at most $O(\sqrt{\log k} \min(|A|, |K - A|))$ and we can use the Hypercontractive Inequality to establish that H is a super-constant edge expander on any set A of size $o(|K|)$. So the ratio of the minimum cut in G to the cut in H is $o(\sqrt{\log k})$ for sets A of size $o(|K|)$, and we can reduce upper-bounding the quality of H to analyzing approximately-balanced cuts. Yet we can apply Bourgain's Junta Theorem, along with the Fourier-theoretic characterization of the cut function of H, to show that for any approximately-balanced cut $(A, K - A)$, either the cut is close to a Junta and so the minimum cut in G is much smaller than $O(\sqrt{\log k} \min(|A|, |K - A|))$, or the function f_A has a significant Fourier tail wh ich will imply that the cut function of H evaluated on A is $\omega(\min(|A|, |K - A|))$. This would establish the desired quality upper-bound for H, and we defer details to the full version of our paper.

V. Improved Constructions via Lifting

In this section we give a polynomial time construction for a flow-sparsifier that achieves quality at most the quality of

the best flow-sparsifier that can be realized as a distribution over 0-extensions. Consequently this is a construction for flow-sparsifiers (and thus also cut-sparsifiers) that achieve quality $O(\frac{\log k}{\log \log k})$. Given that the current best upper bounds on the quality of flow and cut-sparsifiers are both achieved through contractions, the constructive results we present here match the best known *existential* bounds on quality. All previous constructions [18], [14] need to sacrifice some super-constant factor in order to actually construct cut or flow-sparsifiers.

Our technique, we believe, is of independent interest: we perform a lifting operation on an appropriate linear program. This lifting operation generates a set of constraints on H that is both stronger than the constraint that H be a flow-sparsifier, and yet weaker than the constraint that H can be written as a convex combination of 0-extension graphs G_f. On one hand, designing a separation oracle becomes easier; while on the other hand, we can reasonably expect to describe the set of feasible graphs H using only polynomially many variables.

Theorem 5. *Given an instance $\mathcal{H} = (G, K)$, there is a polynomial (in n and k) time algorithm that outputs a flow sparsifier H of quality $\alpha \leq \alpha'(\mathcal{H})$, where $\alpha'(\mathcal{H})$ is the Contraction Quality of G, K.*

Proof: We show that the following LP can give a flow-sparsifier with the desired properties:

$$\min \quad \alpha$$

s.t

$$
\begin{aligned}
cong_G(\vec{w}) &\leq \alpha \\
w_{i,j} &= \sum_{u,v \in V, u \neq v} c(u,v) x_{i,j}^{u,v} & \forall i,j \in K \\
x_{i,j}^{u,v} &= x_{j,i}^{v,u} & \forall u,v \in V, u \neq v, i,j \in K \\
\sum_{j \in K} x_{i,j}^{u,v} &= x_i^u & \forall u,v \in V, u \neq v, i \in K \\
\sum_{i \in K} x_i^u &= 1 & \forall u \in V \\
x_{i,i}^i &= 1 & \forall i \in K \\
x_{i,j}^{u,v} &\geq 0 & \forall u,v \in V, u \neq v, i,j \in K
\end{aligned}
$$

Lemma 3. *The value of the LP is at most $\alpha'(\mathcal{H})$.*

Proof: Suppose γ is a distribution on 0-extensions s.t. $H = \sum_f \gamma(f) G_f$ has quality at most $\alpha'(\mathcal{H})$. If we set $x_{i,j}^{u,v}$ equal to the probability that $f(u) = i$ and $f(v) = j$ (when f is sampled from γ), and set the remaining allocation variables in the LP accordingly, this will define a feasible solution to the LP with $\alpha \leq \alpha'(\mathcal{H})$. ∎

There are qualitatively two types of constraints that are associated with good flow-sparsifiers H: All flows routable in H with congestion at most 1 must be routable in G with congestion at most α. This constraint is *equivalent* to the constraint that $\vec{H}$ can be routed in G with congestion at most α.

The second set of constraints associated with good flow-

sparsifiers are that all flows routable in G with congestion at most 1 can also be routed in H with congestion at most 1. This constraint can also be written as an infinite number of linear constraints on H, but no polynomial time separation oracle is known for these constraints. Instead, previous work relied on using oblivious routing guarantees to get an approximate separation oracle for this problem.

The above linear program actually implicitly enforces this constraint! Let the edge capacities in H be defined by $\{w_{i,j} : i, j \in K, i < j\}$ for a feasible solution to the above LP.

Lemma 4. *H is a flow-sparsifier*

Proof: Let $\{f_{i,j} : i, j \in K, i < j\}$ be a multicommodity flow that can be routed in G. By the LP duality, we have $\sum_{u<v} c(u,v) \delta(u,v) \geq \sum_{i<j} f_{i,j} \delta(i,j)$ for every metric δ over V. Let EMD_δ be the Earth-mover distance between two distributions according to the metric δ. Let δ' be an arbitrary metric over K. Then we can re-write $\sum_{i<j} \delta'(i,j) w_{i,j}$ as

$$\sum_{u<v} c(u,v) \sum_{i \neq j} x_{i,j}^{u,v} \delta'(i,j) \geq \sum_{u<v} c(u,v) EMD_{\delta'}(x^u, x^v).$$

Define $\delta(u,v) = EMD_{\delta'}(x^u, x^v)$. Clearly, δ is a metric over V and $\delta(i,j) = \delta'(i,j)$ for every $i,j \in K$. Because f can be routed in G, we have

$$\sum_{u<v} c(u,v) \delta(u,v) \geq \sum_{i<j} f_{i,j} \delta(i,j) = \sum_{i<j} f_{i,j} \delta'(i,j)$$

Directly from the definition of H, we have the condition that $\sum_{u<v} c(u,v) \sum_{i \neq j} x_{i,j}^{u,v} \delta'(i,j) = \sum_{i<j} \delta'(i,j) w_{i,j}$. Since δ' is an arbitrary metric over K, by strong duality f can also be routed in H. ∎

Of course the LP explicitly enforces the constraint that H can be routed in G with congestion at most α, and hence H has quality at most α. Also this LP can be solved in polynomial time, and so this implies the theorem. ∎

VI. Abstract Integrality Gaps and Rounding Algorithms

In this section, we give a generalization of the hierarchical decompositions constructed in [20]. This immediately yields an $O(\log k)$-competitive Steiner oblivious routing scheme, which is optimal. Also, from our hierarchical decompositions we can recover the $O(\log k)$ bound on the flow-cut gap for maximum concurrent flows given in [16] and [1]. Additionally, we can also give an $O(\log k)$ flow-cut gap for the maximum multiflow problem, which was originally given in [10]. This even yields an $O(\log k)$ flow-cut gap for the relaxation for the requirement cut problem, which is given in [19]. In fact, we will be able to give an abstract framework to which the results in this section apply (and yield $O(\log k)$ flow-cut gaps for), and in this sense we are able to help explain why $O(\log k)$ is often the worst-case ratio of the optimal integral cover compared to optimal fractional packing in undirected graphs.

Philosophically, this section aims to answer the question: Do we really need to pay a price in the approximation guarantee for reducing to a graph on size k? In fact, as we will see, there is often a way to combine both the reduction to a graph of size k and the rounding algorithm needed to bound the flow-cut gap into one step!

A. 0-Decomposition

We extend the notion of 0-extensions to a notion of 0-decompositions. Intuitively, we would like to combine the notion of a 0-extension with that of a decomposition tree.

Again, given a 0-extension f, we will denote G_f as the graph on K that results from contracting all sets of nodes mapped to any single terminal. Then we will use c_f to denote the capacity function of this graph.

Definition 11. *Given a tree T on K, and a 0-extension f, we can generate a 0-decomposition $G_{f,T} = (K, E_{f,T})$ as follows:*

The only edges present in $G_{f,T}$ will be those in T, and for any edge $(a,b) \in E(T)$, let T_a, T_b be the subtrees containing a, b respectively that result from deleting (a,b) from T.

Then $c_{f,T}(a,b)$ (i.e. the capacity assigned to (a,b) in $G_{f,T}$ is: $c_{f,T}(a,b) = \sum_{u,v \in K}$ and $_{u \in T_a, v \in T_b} c_f(u,v)$.

Let Λ denote the set of 0-extensions, and let Π denote the set of trees on K.

Claim 3. *For any distribution γ on $\Lambda \times \Pi$, and for any demand $\vec{d} \in \mathfrak{R}^{\binom{K}{2}}$, $cong_H(\vec{d}) \leq cong_G(\vec{d})$ where $H = \sum_{f \in \Lambda, T \in \Pi} \gamma(f,T) G_{f,T}$*

Proof: Clearly for all f, T, $\gamma(f,T)\vec{d}$ is feasible in $\gamma(f,T)G_f$ (because contracting edges only makes routing flow easier), and so because $G_{f,T}$ is a hierarchical decomposition tree for G_f, then it follows that $\gamma(f,T)\vec{d}$ is also feasible in $G_{f,T}$. ∎

Claim 4. *Given any distribution γ on $\Lambda \times \Pi$, let $H = \sum_{f \in \Lambda, T \in \Pi} \gamma(f,T) G_{f,T}$. Then $\sup_{\vec{d} \in \mathfrak{R}^{\binom{K}{2}}} \frac{cong_G(\vec{d})}{cong_H(\vec{d})} = cong_G(\vec{H})$*

Theorem 6. *There is a polynomial time algorithm to construct a distribution γ on $\Lambda \times \Pi$ such that $cong_G(\vec{H}) = O(\log k)$ where $H = \sum_{f \in \Lambda, T \in \Pi} \gamma(f,T) G_{f,T}$.*

We want to show that there is a distribution γ on $\Lambda \times \Pi$ such that $cong_G(\vec{H}) = O(\log k)$. This will yield a generalization of the results in [20]. In order to prove such a distribution exists, we follow the plan given in [18] and [14]: we set up a zero-sum game in such a way that a bound of $O(\log k)$ on the game value will imply our desired structural result.

This is precisely how (similar) zero-sum games are used in [18] and [14] to prove that good cut-sparsifiers and flow-sparsifiers exist, respectively. We defer the precise description of the game, and the analysis to the full version of the paper but we want to highlight that the main difference is that while previous work constructed good responses (and hence

bounds on the game value) based on rounding algorithms due to [8] and [5] for the 0-extension problem, here we slightly modify the rounding algorithm due to [9] in order to generate good "tree-like" responses.

B. Applications

Also, as we noted, this gives us an alternate proof of the main results in [16], [1] and [10]. We first give an abstract framework into which these problems all fit:

Definition 1. We call a fractional packing problem P a graph packing problem if the goal of the dual covering problem D is to minimize the ratio of the total units of distance × capacity allocated in the graph divided by some monotone increasing function of the distances between terminals.

Let ID denote the integral dual graph covering problem. We can formally define ID by a family $\mathcal{D}$ of admissible metrics over the terminals K. Then the goal of ID is to find a metric $\delta \in \mathcal{D}$ and a 0-extension $f : V \to K$ such that the ratio of δ_f × capacity divided by the monotone function applied to δ_f is minimized. Here, δ_f is a metric over V such that $\delta_f(u,v) = \delta(f(u), f(v))$. To make this definition seem more natural, we demonstrate that a number of well-studied problems fit into this framework.

Example 1. *[15], [16], [1] P: maximum concurrent flow; ID: generalized sparsest cut*

Here we are given some demand vector $\vec{f} \in \mathfrak{R}^{\binom{K}{2}}$, and the goal is to maximize the value r such that $r\vec{f}$ is feasible in G. Then the dual to this problem corresponds to minimizing the total distance × capacity units, divided by $\sum_{(a,b)} \vec{f}_{a,b} d(a,b)$, where d is the induced semi-metric on K. The function in the denominator is clearly a monotone increasing function of the distances between pairs of terminals, and hence is an example of what we call a graph packing problem. The generalized sparsest cut problem corresponds to the "integral" constraint on the dual, that the distance function be a cut metric.

Example 2. *[10] P: maximum multiflow; ID: multicut*

Here we are given some pairs of terminals $T \subset \binom{K}{2}$, and the goal is to find a flow $\vec{f}$ that can be routed in G that maximizes $\sum_{(a,b) \in T} \vec{f}_{a,b}$. The dual to this problem corresponds to minimizing the total distance × capacity units divided by $\min_{(a,b) \in T} \{d(a,b)\}$, again where where d is the induced semi-metric on K. Also the function in the denominator is again a monotone increasing function of the distances between pairs of terminals, and hence is another an example of what we call a graph packing problem. The multicut problem corresponds to the "integral" constraint on the dual that the distance function be a partition metric.

Example 3. *ID: Steiner multi-cut*

Example 4. *ID: Steiner minimum-bisection*

Example 5. *[19] P: multicast routing; ID: requirement cut*

This is another partitioning problem, and the input is again a set of subsets $\{R_i\}_i$. Each subset R_i is also given at requirement r_i, and the goal is to minimize the total capacity removed from G, in order to ensure that each subset R_i is contained in at least r_i different components. Similarly to the Steiner multi-cut problem, the standard relaxation for this problem is to minimize the total amount of distance $\times$ capacity units allocated in G, s.t. for each i the minimum spanning tree T_i (on the induced metric on K) on every subset R_i has total distance at least r_i. Let Π_i be the set of spanning trees on the subset R_i. Then we can again cast this relaxation in the above framework because the goal is to minimize the total distance $\times$ capacity units divided by $\min_i\{\frac{\min_{T\in\Pi_i}\sum_{(a,b)\in T}d(a,b)}{r_i}\}$. The dual to this fractional covering problem is actually a common encoding of multicast routing problems, and so these problems as well are examples of graph packing problems. Here the requirement cut problem corresponds to the "integral" constraint that the distance function be a partition metric.

In fact, one could imagine many other examples of interesting problems that fit into this framework. One can regard maximum multiflow as an *unrooted* problem of packing an edge fractionally into a graph G, and the maximum concurrent flow problem is a rooted graph packing problem where we are given a fixed graph on the terminals (corresponding to the demand graph) and the goal is to pack as many copies as we can into G (i.e. maximizing throughput). The dual to the Steiner multi-cut is more interesting, and is actually a combination of rooted and unrooted problems where we are given subset R_i of terminals, and the goal is to maximize the total spanning trees over the sets R_i that we pack into G. This is a combination of a unrooted (each spanning tree on any set R_i counts the same) and a rooted problem (once we fix the R_i, we need a spanning tree on these terminals). Then any other flow-problem that is combinatorially restricted can also be seen to fit into this framework.

Using Theorem 6, we prove:

Theorem 4. There is a polynomial time algorithm to construct a distribution μ on (a polynomial number of) trees on the terminal set K, s.t.

$$E_{T\leftarrow\mu}[OPT(P,T)] \leq O(\log k)OPT(P,G)$$

and such that any valid integral dual of cost C (for any tree T in the support of μ) can be immediately transformed into a valid integral dual in G of cost at most C.

We first demonstrate that the operations we need to construct a 0-decomposition only make the dual to a graph packing problem more difficult: Let $v(G,K)$ be the optimal value of a dual to a graph packing problem on $G = (V,E)$, $K \subset V$.

Claim 5. *Replacing any edge (u,v) of capacity $c(u,v)$ with a path $u = p_1, p_2, ..., p_r = v$, deleting the edge (u,v) and adding $c(u,v)$ units of capacity along the path cannot decrease the optimal value of the dual.*

Proof: We can scale the distance function of the optimal dual so that the monotone increasing function of the distances between terminals is exactly 1. Then the value of the dual is exactly the total capacity $\times$ distance units allocated. If we maintain the same metric space on the vertex set V, then the monotone increasing function of terminal distances is still exactly 1 after replacing the edge (u,v) by the path $u = p_1, p_2, ..., p_r = v$. However this replacement does change the cost (in terms of the total distance $\times$ capacity units). Deleting the edge reduces the cost by $c(u,v)d(u,v)$, and augmenting along the path increases the cost by $c(u,v)\sum_{i=1}^{r-1}d(p_i, p_{i+1})$ which, using the triangle inequality, is at least $c(u,v)d(u,v)$. ∎

Claim 6. *Suppose we contract two nodes u,v (s.t. not both of u,v are terminals) - then the optimal value of the dual does not decrease.*

Proof: We can equivalently regard this operation as placing an edge of infinite capacity connecting u and v, and this operation clearly does not change the set of distance functions for which the monotone increasing function of the terminal distances is at least 1. And so this operation can only increase the cost of the optimal dual solution. ∎

We can obtain any 0-decomposition $G_{f,T}$ from some combination of these operations. So we get that for any f, T:

Corollary 1. $v(G_{f,T}, K) \geq v(G,K)$

Let γ be the distribution on $\Lambda \times \Pi$ s.t. $H = \sum_{f\in\Lambda, T\in\Pi}\gamma(f,T)G_{f,T}$ and $cong_G(\vec{H}) \leq O(\log k)$.

Lemma 5. $E_{(f,T)\leftarrow\gamma}[v(G_{f,T}, K)] \leq O(\log k)v(G,K)$.

Proof: We know that there is a metric d on V s.t. $\sum_{(u,v)}c(u,v)d(u,v) = v(G,K)$ and that the monotone increasing function of d (restricted to K) is at least 1.

We also know that there is a simultaneous routing of each $\gamma(f,T)G_{f,T}$ in G so that the congestion on any edge in G is $O(\log k)$. Then consider the routing of one such $\gamma(f,T)G_{f,T}$ in this simultaneous routing. Each edge $(a,b) \in E_{f,T}$ is routed to some distribution on paths connecting a and b in G. In total $\gamma(f,T)c_f(a,b)$ flow is routed on some distribution on paths, and consider a path p that carries $C(p)$ total flow from a to b in the routing of $\gamma(f,T)G_{f,T}$. If the total distance along this path is $d(p)$, we increment the distance $d_{f,T}$ on the edge (a,b) in $G_{f,T}$ by $\frac{d(p)C(p)}{\gamma(f,T)c_{f,T}(a,b)}$, and we do this for all such paths. We do this also for each (a,b) in $G_{f,T}$.

If $d_{f,T}$ is the resulting semi-metric on $G_{f,T}$, then this distance function dominates d restricted to K, because the distance that we allocate to the edge (a,b) in $G_{f,T}$ is a convex combination of the distances along paths connecting a and

b in G, each of which is at least $d(a,b)$.

So if we perform the above distance allocation for each $G_{f,T}$, then each resulting $d_{f,T}, G_{f,T}$ pair satisfies the condition that the monotone increasing function of terminal distances $(d_{f,T})$ is at least 1. But how much distance $\times$ capacity units have we allocated in expectation?

$$E_{(f,T)\leftarrow\gamma}[\nu(G_{f,T}, K)] \leq \sum_{f,T} \gamma(f,T) \sum_{(a,b)\in E_{f,T}} c_{f,T}(a,b) d_{f,T}(a,b)$$

We can re-write the above double sum as:

$$\sum_{(a,b)\in E} flow_{\vec{H}}(e) d(a,b) \leq cong_G(\vec{H}) \sum_{(a,b)\in E} c(a,b) d(a,b)$$
$$\leq O(\log k)\nu(G,K)$$

■

And this implies:

Theorem 3. For any graph packing problem P, the maximum ratio of the integral dual to the fractional primal is at most $O(\log k)$ times the maximum ratio restricted to trees.

And since we can actually construct such a distribution on 0-decompositions in polynomial time using Theorem 6, we obtain Theorem 4 which immediately implies that given an approximation algorithm that achieves an approximation ratio of C for the integral dual to a graph packing problem on trees, we obtain an approximation algorithm with a guarantee of $O(C \log k)$ for general graphs.

For example, this gives a generic algorithm that achieves an $O(\log k)$ guarantee for *both* generalized sparsest cut and multicut. The previous techniques for rounding a fractional solution to generalized sparsest cut [16], [1] rely on metric embedding results, and the techniques for rounding fractional solutions to multicut [10] rely on purely combinatorial, region-growing arguments. Yet, through this theorem, we can give a unified rounding algorithm that achieves an $O(\log k)$ guarantee for both of these problems, and more generally for graph packing problems (whenever the integrality gap restricted to trees is a constant).

OPEN PROBLEMS

The main open problem is whether there exist $\tilde{O}(\sqrt{\log k})$-quality cut-sparsifiers. In fact, there could be even better quality cut-sparsifiers in general that beat the "0-Extension Bound" in [18] and achieve quality $o(\sqrt{\log k})$, which would be a truly surprising result.

REFERENCES

[1] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SICOMP*, pages 291–301, 1998.

[2] W. Beckner. Inequalities in fourier analysis. *Annals of Mathematics*, pages 159–182, 1975.

[3] A. Bonami. Etude des coefficients de foureir des fonctions de $l^p(g)$. *Annales de "institut fourier*, pages 335–402, 1970.

[4] J. Bourgain. On the distribution of the fourier spectrum of boolean functions. *Israel Journal of Mathematics*, pages 269–276, 2002.

[5] G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. *SODA*, pages 8–16, 2001.

[6] I. Dinur, E. Friedgut, G. Kindler, and R. O'Donnell. On the fourier tails of bounded functions over the discrete cube. *Israel Journal of Mathematics*, pages 389–412, 2007.

[7] M. Englert, A. Gupta, R. Krauthgamer, H. Räcke, I. Talgam-Cohen, and K. Talwar. Vertex sparsifiers: new results from old techniques *APPROX*, 2010, to appear.

[8] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar. An improved approximation algorithm for the 0-extension problem. *SODA*, pages 257–265, 2003.

[9] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *STOC*, pages 448–455, 2003.

[10] N. Garg, V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SICOMP*, 25:235–251, 1996.

[11] W. Johnson, J. Lindenstrauss, and G. Schechtman. Extensions of lipschitz maps into banach spaces. *Israel Journal of Mathematics*, pages 129–138, 1986.

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

[13] S. Khot and A. Naor. Nonembeddability theorems via fourier analysis. *FOCS*, pages 101–112, 2005.

[14] T. Leighton and A. Moitra. Extensions and limits to vertex sparsification. *STOC*, pages 47–56, 2010.

[15] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *JACM*, pages 787–832, 1999.

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

[17] K. Makarychev and Y. Makarychev. Metric extension operators, vertex sparsifiers and lipschitz extendability. *FOCS*, 2010, this proceedings.

[18] A. Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. *FOCS*, pages 3–12, 2009.

[19] V. Nagarajan and R. Ravi. Approximation algorithms for requirement cut on graphs. *APPROX*, pages 209–220, 2005.

[20] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. *STOC*, pages 255–264, 2008.