Extensions and Limits to Vertex Sparsification

Tom Leighton [*] Ankur Moitra [†]

Abstract

Suppose we are given a graph $G = (V, E)$ and a set of terminals $K \subset V$. We consider the problem of constructing a graph $H = (K, E_H)$ that approximately preserves the congestion of every multicommodity flow with endpoints supported in K. We refer to such a graph as a *flow sparsifier*. We prove that there exist flow sparsifiers that simultaneously preserve the congestion of all multicommodity flows within an $O(\log k / \log\log k)$-factor where $|K| = k$. This bound improves to $O(1)$ if G excludes any fixed minor. This is a strengthening of previous results, which consider the problem of finding a graph $H = (K, E_H)$ (a *cut sparsifier*) that approximately preserves the value of minimum cuts separating any partition of the terminals. Indirectly our result also allows us to give a construction for better quality cut sparsifiers (and flow sparsifiers). Thereby, we immediately improve all approximation ratios derived using vertex sparsification in [22].

We also prove an $\Omega(\log\log k)$ lower bound for how well a flow sparsifier can simultaneously approximate the congestion of every multicommodity flow in the original graph. Our proof crucially relies on a geometric phenomenon pertaining to the unit congestion polytope of an expander graph, which we exploit in order to prove sub-optimal but super-constant congestion lower bounds against many multicommodity flows at once. Our result implies that approximation algorithms for multicommodity flow-type problems designed by a black box reduction to a "uniform" case on k nodes (see [22] for examples) must incur a super-constant cost in the approximation ratio.

[*] ftl@math.mit.edu Massachusetts Institute of Technology and Akamai Technologies, Inc

[†] moitra@mit.edu Massachusetts Institute of Technology. This research was supported in part by a Fannie and John Hertz Foundation Fellowship. Part of this work was done while the author was an intern at Microsoft Research New England

1 Introduction

Suppose we are given an n-node graph $G = (V, E)$ and a set of terminals $K \subset V$. We consider the problem of constructing a graph $H = (K, E_H)$ on just the terminal set K that approximately preserves the congestion of every multicommodity flow with endpoints supported in K. We refer to such a graph as a *flow sparsifier*.

1.1 Background

The notion of exactly representing or approximately preserving certain combinatorial properties of a graph G on a simpler graph is ubiquitous in combinatorial optimization, and has many applications in theoretical computer science (since such representations are often used to design faster and/or better approximation algorithms). For example, Mader's theorem implies that if we are only interested in the pair-wise minimum cuts between terminals - i.e. for all $a, b \in K$, the minimum cut separating a and b in G - then there is a capacitated graph $H = (K, E_H)$ that preserves all such cuts *exactly*. Additionally, elegant generalizations of Mader's theorem are given in [9], and these results are applied to some problems in network design which require subgraph solutions to be simple, or are subject to degree constraints.

Benczúr and Karger [5] proved that there is a capacitated graph $H = (V, E_H)$ on $O(n \log n)$ edges that approximates *all cuts* in $G = (V, E)$ to within a $1 + \epsilon$ factor, and gave a construction for such graphs that runs in nearly linear time. These results are now a common pre-processing step used to speed up the run-time of many algorithms and approximation algorithms (e.g. [5], [18]). Additionally, Chew [10] introduced the notion of a spanner - a sparse graph that approximates the shortest path distances in the original graph - and gave applications to motion planning in the plane. Related notions of spanners have long been studied in access control, property testing and data structures (see [6]). Spielman and Teng [25] gave a nearly-linear time algorithm for constructing weighted graphs on a nearly-linear number of edges that approximately preserve the Laplacian. These results are improved in [24] and in [4], and approximately preserving the Laplacian is an important step in devising nearly-linear time algorithms for solving diagonally dominant systems of linear equations [25].

Moitra [22] considered the problem of constructing a graph $H = (K, E_H)$ that approximately preserves the values of minimum cuts separating *any* partition of the terminals. We refer to such a graph as a *cut sparsifier*. Approximation algorithms can be run on H as a proxy for the original graph, thereby yielding the first $poly(\log k)$-approximation algorithms (or competitive ratios) for cut and flow problems such as l-multicut, requirement cut, various linear arrangement problems and Steiner oblivious routing. These results give approximation guarantees (or competitive ratios) that are independent of the size of the underlying graph, and only depend polylogarithmically on the number of "interesting" nodes.

1.2 Our Results

In this paper we strengthen the results in [22] by finding a graph $H = (K, E_H)$ (that we call a flow sparsifier) that approximately preserves the congestion of every multicommodity flows with endpoints supported in K (rather than just requiring the value of minimum cuts separating any partition of the terminals to be preserved). Flow sparsifiers can be obtained from cut sparsifiers, with some degradation in quality corresponding to how well multicommodity flows are approximated by sparsest cuts. Here we consider flow sparsification directly, and are able to achieve the same quality guarantee for this stronger condition on flows as the previous work achieved for the weaker condition on just cuts.

1.2.1 The Quality of a Flow Sparsifier

Throughout this paper, we will consider only demand vectors whose endpoints are supported in the set K.

Given a graph $G = (V, E)$ and a set of terminals $K \subset V$, we say that a graph $H = (K, E_H)$ is a *flow sparsifier* if every demand vector that is feasible in G is also feasible in H. Then we define the *quality* of a flow sparsifier H as the worst-case ratio of the congestion of a flow in G to the congestion of the flow in H. So the quality of a flow sparsifier represents how well H approximates G as a flow network. In Section 3, we prove:

Theorem 1. *For any capacitated graph $G = (V, E)$ and any set $K \subset V$ of size k, there is an $O(\log k / \log\log k)$-quality flow sparsifier $H = (K, E_H)$. If G excludes a fixed minor, then there is an $O(1)$-quality flow sparsifier.*

In Section 2.2 we demonstrate that this notion of a flow sparsifier is stronger than the notion of a cut sparsifier in [22]. Following the method in [22], we prove this existential result by analyzing a zero-sum game. We note that in [22], the question of bounding the game value is transformed to a question about the integrality gap of the linear program [8] for the 0-extension problem. Yet only ℓ_1-metrics arise as metric costs in analyzing this game. But by allowing more expressive metric spaces, we are able to encode a game that attempts to preserve the congestion of multicommodity flows, which allows us to prove stronger results.

From this, we can directly improve the competitive ratio of Steiner oblivious routing by an $O(\log k)$ factor over previous results. With further effort, this result also allows us to write a stronger linear program for the problem of actually constructing a good flow sparsifier. Consequently, we can simplify the approximate separation oracles in [22] and give a construction for better quality flow sparsifiers:

Theorem 2. *For any capacitated graph $G = (V, E)$ and any set $K \subset V$ of size k, there is a polynomial (in n and k) time algorithm to construct an $O(\log^2 k / \log\log k)$-quality flow sparsifier $H = (K, E_H)$.*

The quality of these flow sparsifiers improves upon the quality of the cut sparsifiers in previous results by a factor of $\tilde{O}(\log^{1.5} k)$. Thereby, we can immediately improve the approximation ratios given in [22] for requirement-cut, l-multicut, and generalizations of min-cut linear arrangement and minimum linear arrangement by a factor of $\tilde{O}(\log^{1.5} k)$. Independently, Gupta, Nagarajan, and Ravi [14] have given an approximation algorithm for requirement cut that is also independent of the size of the graph, and improves upon the approximation algorithm implied by this paper.

1.2.2 Limits to Flow Sparsification

As we have noted above, many combinatorial properties of a graph G can be represented exactly or approximated within a constant factor on a smaller graph. In Section 6 we prove that the congestion of each multicommodity flow (with endpoints supported in K) is a rare example of a combinatorial property that *cannot* be approximated within a constant factor on a smaller graph. In particular, our main result is:

Theorem 3. *For infinitely many values of k, there is a graph $G = (V, E)$ and a set $K \subset V$ of size k for which any flow sparsifier has quality at least $\Omega(\log\log k)$*

Our proof crucially relies on a geometric phenomenon pertaining to the unit congestion polytope of an expander graph (we explain this in more detail in Section 5), which we exploit in order to prove sub-optimal but super-constant congestion lower bounds against many multicommodity flows at once.

Despite the long list of results on multicommodity flows, sparsest cut, and discrete metric spaces, no super-constant lower bound on the quality of a flow sparsifier was known prior to this work. We consider this a fundamental and natural question in the study of multicommodity flows. Our result implies that approximation algorithms for multicommodity flow-type problems designed by a black box reduction to a "uniform" case on k nodes (see [22] for examples) must incur a super-constant cost in the approximation ratio.

1.3 Network Coding

Here we briefly describe a somewhat surprising connection of this work to network coding theory. In this paper, we prove that there are $O(\log k / \log\log k)$-quality flow sparsifiers. And in fact, these flow sparsifiers are generated as a convex combination of 0-extensions. Any such flow sparsifier preserves not only the multicommodity flow rate, but also the network coding rate (see [15] for a definition):

Theorem 4. *Any flow sparsifier generated as a convex combination of* 0*-extensions that has quality* ν *also preserves the rates of all network coding problems with endpoints supported in* K *to within a* ν*-factor.*

It is conjectured [1], [15], [21] that in undirected graphs, the network coding rate is always equal to the multicommodity flow rate. This conjecture would resolve a long-standing open question of Floyd [12] and Aggarwal and Vitter [2], and in fact would give the first unrestricted super-linear lower bounds in the cell-probe model for computation [1] *for any problem*. In general graphs, we know nothing about this conjecture, yet for the existential results on flow sparsifiers that we present here, we automatically preserve the network coding rate at least as well as the multicommodity flow rate. These results are given in Section 7.

2 Maximum Concurrent Flow

An instance of the maximum concurrent flow problem consists of an undirected graph $G = (V, E)$, a capacity function $c : E \to \mathfrak{R}^+$ 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 :

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

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 $\vec{f} \in \mathfrak{R}^{\binom{k}{2}}$ 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 $s_i = x, t_i = y$.

Definition 1. *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).

2.1 Flow Sparsifiers

Definition 2. $H = (K, E_H)$ *is a flow sparsifier if for all* $\vec{f} \in \mathfrak{R}^{\binom{k}{2}}$, $cong_H(\vec{f}) \leq cong_G(\vec{f})$

So a vertex sparisifer is a "better" flow network than the original graph, in the sense that all demands (with endpoints supported in K) that are routable in G (with congestion at most 1) can also be routed in H.

Definition 3. *The quality of a flow sparsifier H is* $\max_{\vec{f} \in \Re^{\binom{k}{2}}} \frac{cong_G(\vec{f})}{cong_H(\vec{f})}$

We also make a simple, but very useful observation that if we are given $H = (K, E_H)$ such that H is a flow sparsifier for $G = (V, E)$ and $K \subset V$, then there is a well-defined notion of a hardest flow (feasible in H) to route in G. So we can express the quality of H as a flow sparsifier as a single maximum concurrent flow problem:

Definition 4. *Let $H = (K, E_H)$ be a capacitated graph. Then let $\vec{H} \in \Re^{\binom{k}{2}}$ be the demand vector in which each coordinate (which corresponds to a pair $a, b \in K$) is set to $c_H(a, b)$ - i.e. the capacity of the edge $(a, b) \in E_H$ (if it exists, and is zero if not).*

Claim 1. *If $H = (K, E_H)$ is a flow sparsifier for $G = (V, E)$, $K \subset V$, then the quality of H (as a flow sparsifier) is $cong_G(\vec{H})$.*

Proof: $cong_G(\vec{H})$ is clearly a lower bound for the quality of H as a flow sparsifier, because the flow $\vec{H}$ is feasible in H by saturating all edges. To prove the reverse inequality, given any flow $\vec{f}$ routed in H, we can compose this routing of $\vec{f}$ with the embedding of H into G. If the routing of $\vec{f}$ has congestion at most 1 in H, then composing the routing with the embedding of H into G will result in congestion at most $cong_G(\vec{H})$ in G:

In particular, consider the flow $\vec{f}$ in H, and consider the path decomposition of this flow. If $\vec{f}$ assigns δ units of flow to a path $P_{a,b}$ in H, 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 $c_H(p_i, p_{i+1})$ is α in H. Then for each flow path P connecting p_i to p_{i+1} in the low-congestion routing of H 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. If we consider any edge in H, every flow path in $\vec{f}$ that traverses this edge uses exactly the paths in G to which (a, b) is mapped in the embedding of H into G. The total flow through any edge (a, b) in H is at most $c_H(a, b)$ because $\vec{f}$ is feasible in H, and so we have not scaled up the amount of flow transported on each path in an optimal routing the (demand) graph H into G at all. Consequently the congestion of this (sub-optimal) routing of $\vec{f}$ in G is at most $cong_G(\vec{H})$. □

So this allows us to restate our main theorem as:

Theorem 3. For infinitely many values of k, there is a graph $G = (V, E)$ and $K \subset V$ of size k for which any flow sparsifier $H = (K, E_H)$ has $cong_G(\vec{H}) = \Omega(\log \log k)$

2.2 Flow Sparsifiers are Stronger than Cut Sparsifiers

Here we prove that this notion of a flow sparsifier is stronger than the notion of a cut sparsifier in [22]. : Suppose we are given an undirected, capacitated graph $G = (V, E)$ and a set $K \subset V$ of size k. Let $h : 2^V \to \Re^+$ denote the cut function of G. We define the function $h_K : 2^K \to \Re^+$ 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.

Theorem 5. *[22] There is an undirected, capacitated graph $G' = (K, E')$* on just the terminals *so that the cut function of this graph $h' : 2^K \to \Re^+$ satisfies for all $U \subset K$:*

$$h_K(U) \leq h'(U) \leq O(\log k / \log \log k) h_K(U)$$

Theorem 6. *[22] There is a polynomial (in n and k) time algorithm to construct an undirected, capacitated graph* $G' = (K, E')$ on just the terminals *so that the cut function of this graph* $h' : 2^K \to \Re^+$ *satisfies for all* $U \subset K$*:*

$$h_K(U) \leq h'(U) \leq O(\log^{3.5} k) h_K(U)$$

In particular, we prove that if G' is an α-quality flow sparsifier and if h' is the cut function of G', then $h_K(U) \leq h'(U) \leq \alpha h_K(U)$ for all $U \subset K$. So the existential and constructive results that we present here are a strengthening of previous results. In fact, the quality of these flow sparsifiers that we construct improves upon the quality of the cut sparsifiers that can be constructed using previous results by a factor of $\tilde{O}(\log^{1.5} k)$ in both the general case and the case in which G excludes a fixed minor. So suppose that G' is an α-quality flow sparsifier:

Claim 2. *For all* $A \subset K$*:* $h_K(A) \leq h'(A)$

Proof: Suppose for some $A \subset K$: $h_K(A) > h'(A)$. Then the min-cut max-flow theorem implies that there is a flow $\vec{r}$ feasible in G such that the total demand crossing the (demand) cut $(A, K - A)$ is exactly $h_K(A)$. But this flow $\vec{r}$ cannot be feasible in G' because the cut $(A, K - A)$ has capacity $h'(A)$ in G' which is strictly smaller than the demand crossing the cut. So this implies $cong_G(\vec{r}) \leq 1 < cong_{G'}(\vec{r})$ which is a contradiction. □

Claim 3. *For all* $A \subset K$*:* $h'(A) \leq \alpha h_K(A)$

Proof: Consider the flow $\vec{G'}$. This flow is feasible in G' - i.e. $cong_{G'}(\vec{G'}) = 1$. Now suppose that there is a cut $A \subset K$ such that $h'(A) > \alpha h_K(A)$. Choose the set $U \subset V$, $U \cap K = A$ such that $h(U) = h_K(A)$. The capacity crossing the cut $(U, V - U)$ is $h_K(A)$ but the total demand crossing the cut is $h'(A)$ so

$$cong_G(\vec{G'}) \geq \frac{h'(A)}{h_K(A)} > \alpha$$

and $cong_G(\vec{G'}) > \alpha cong_{G'}(\vec{G'})$ which is a contradiction. □

3 Good Flow Sparsifiers Exist

We prove that there exist flow sparsifiers that preserve the congestion of all multicommodity flows within an $O(\log k / \log\log k)$-factor, and this bound improves to $O(1)$ if G excludes any fixed minor. This is a strengthening of previous results, which only guarantee that the value of minimum cuts separating any partition of the terminals is preserved. We prove this through a zero-sum game between an extension player and a congestion player. This game is similar to the one introduced in [22], but because the game will be played over the space of multicommodity flows we will need a more intricate argument.

3.1 0-Extensions

The 0-extension problem was originally formulated by Karzanov [17] who introduced the problem as a natural generalization of the minimum multiway cut problem [17]. Suppose we are given an undirected, capacitated graph $G = (V, E)$, $c : E \to \Re^+$, a set of terminals $K \subset V$ and a metric space 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:

Definition 5. $f : V \to K$ *is a* 0*-extension if for all* $a \in K$, $f(a) = a$.

Definition 6. *Given a graph $G = (V, E)$ and a set $K \subset V$, and a 0-extension f, $G_f = (K, E_f)$ is a capacitated graph in which for all $a, b \in K$, the capacity of edge $(a, b) \in E_f$ is*

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

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. And G_f results from collapsing each cluster corresponding to a terminal and preserving all edges joining distinct clusters.

Claim 4. *For any 0-extension f, and for any demand vector $\vec{d} \in \Re^{\binom{k}{2}}$, $cong_{G_f}(\vec{d}) \leq cong_G(\vec{d})$*

Roughly, contracting an edge can only make routing easier. The goal of the 0-extension problem is to find a 0-extension f so that $\sum_{(u,v)\in E} c(u,v)D(f(u), f(v))$ is minimized. When D is just the uniform metric on the terminals K, this exactly the minimum multiway cut problem. Karzanov gave a (semi)metric relaxation of the 0-extension problem [17]:

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

Let OPT^* denote the value of an optimal solution to the above linear programming relaxation of the 0-extension problem. Also let OPT denote the value of an optimal solution to the 0-extension problem. Clearly $OPT^* \leq OPT$. Calinescu, Karloff and Rabani [8] were the first to give bounds on the integrality gap of this linear program for general graphs, and proved an $O(\log k)$ bound. We will make use of an improved bound due to Fakcharoenphol, Harrelson, Rao and Talwar:

Theorem 7. *[11] $OPT \leq O(\frac{\log k}{\log\log k})OPT^*$*

We will use the above theorem to show that, existentially, there is a graph H that that is an $O(\log k/\log\log k)$-quality flow sparsifier. In fact, this graph will be a convex combination of 0-extensions graphs G_f of G.

We will also use a theorem due to Calinescu, Karloff and Rabani which gives a $O(1)$ bound when G excludes a fixed minor:

Theorem 8. *[8] Suppose that G excludes $K_{r,r}$ as a minor. Then $OPT \leq O(r^2)OPT^*$*

3.2 A Zero-Sum Game

We can define the unit congestion polytope P_G with respect to G as:

$$P_G = \{\vec{f} \in \Re^{\binom{k}{2}} | cong_G(\vec{f}) \leq 1\}$$

We will also be interested in the boundary

$$S_G = \{\vec{f} \in \Re^{\binom{k}{2}} | cong_G(\vec{f}) = 1\}$$

Definition 7. *We will call a metric space d on K realizable in G if there is some extension of d to V so that $\sum_{(u,v)\in E} c(u,v)d(u,v) \leq 1$.*

The set D_G of realizable metric spaces is clearly convex.

Claim 5. *The number of vertices in the polytope D_G is finite.*

Proof: To prove this claim, consider the polytope P_G. This polytope can be realized as the projection of a higher dimensional polytope (that actually finds the realization of each multicommodity flow, and uses flow-conservation constraints as opposed to an explicit decomposition into flow paths). This higher dimensional polytope has a (polynomial in n, k) number of constraints, and hence a finite number of vertices and this implies that the projected polytope P_G also has a finite number of vertices. Each distinct facet of P_G contains a distinct subset of the vertices of P_G, and hence the number of facets of P_G is also finite. The facets of P_G are exactly the polar duals to the vertices of D_G, and this implies the claim. □

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-congestion game:

> The extension player (P1) chooses a 0-extension f
> The congestion player (P2) chooses a vertex $d_{\vec{r}}$ of D_G

We write $d_{\vec{r}}$ for a vertex of D_G because we will use $\vec{r}$ to denote a vertex of P_G that the half-space perpendicular to $d_{\vec{r}}$ contains. Equivalently, $\vec{r}$ is a multicommodity flow that is feasible in G and for which $\sum_{(a,b)} \vec{r}_{a,b} d_{\vec{r}}(a,b) = 1$.

Given a strategy f for P1 and a strategy $d_{\vec{r}}$ for P2, P2 wins

$$\begin{aligned} M(f, d_{\vec{r}}) &= \sum_{u,v \in K} d_{\vec{r}}(u,v) c_f(u,v) \\ &= \sum_{(a,b) \in E} d_{\vec{r}}(f(a), f(b)) c(a,b) \end{aligned}$$

where $c_f : K \times K \to \Re^+$ is the capacity function for G_f.

The advantage of restricting the congestion player's strategies to be vertices of D_G is that the joint strategy space of the game is finite, so we can apply the standard Min-Max Theorem for finite strategy space games, rather than more general Min-Max Theorems for arbitrary convex spaces that are not necessarily characterized as the convex hull of a finite number of points.

Definition 8. *Let ν denote the game value of the extension-congestion game*

Lemma 1. $\nu \leq O(\log k / \log\log k)$

Proof: 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 vertices of D_G. We can define a semi-metric on V: $\delta(a,b) = \sum_{d_{\vec{r}}} \mu(d_{\vec{r}}) d_{\vec{r}}(a,b)$. Note that

$$\begin{aligned} \sum_{(a,b) \in E} \delta(a,b) c(a,b) &= \sum_{(a,b) \in E} \sum_{d_{\vec{r}}} \mu(d_{\vec{r}}) d_{\vec{r}}(a,b) c(a,b) \\ &= \sum_{d_{\vec{r}}} \mu(d_{\vec{r}}) \sum_{(a,b) \in E} d_{\vec{r}}(a,b) c(a,b) \\ &= \sum_{d_{\vec{r}}} \mu(d_{\vec{r}}) = 1 \end{aligned}$$

Then using the theorem due to Fakcharoenphol, Harrelson, Rao and Talwar [11], there exists a 0-extension f such that

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

Then suppose P1 plays such a strategy f:

$$\begin{aligned} E_{d_{\vec{r}} \leftarrow \mu}[M(f, d_{\vec{r}})] &= \sum_{d_{\vec{r}}} \mu(d_{\vec{r}}) \sum_{(a,b) \in E} d_{\vec{r}}(f(a), f(b)) c(a,b) \\ &= \sum_{(a,b) \in E} c(a,b) \sum_{d_{\vec{r}}} \mu(d_{\vec{r}}) d_{\vec{r}}(f(a), f(b)) \\ &= \sum_{(a,b) \in E} c(a,b) \delta(f(a), f(b)) \\ &= O(\frac{\log k}{\log \log k}) \end{aligned}$$

□

Note also that we can replace the application of the theorem due to [11] in the zero-sum game above, and use instead the theorem due to [8]. And in the case in which G excludes $K_{r,r}$ as a minor, this gives us an improved bound of $\nu \leq O(r^2)$ on the game value.

Theorem 1. For any capacitated graph $G = (V, E)$ and any set $K \subset V$ of size k, there is an $O(\log k / \log \log k)$-quality flow sparsifier $H = (K, E_H)$. And if G excludes $K_{r,r}$ as a minor, then this bound improves to $O(r^2)$.

Proof: We can again apply von Neumann's Min-Max Theorem, and get that there exists a distribution γ on 0-extensions such that $E_{f \leftarrow \gamma}[M(f, d_{\vec{r}})] = \nu$ for all $d_{\vec{r}} \in D_G$. Then let

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

Let $d_{\vec{r}} \in D_G$ be arbitrary. Then

$$\begin{aligned} E_{f \leftarrow \gamma}[M(f, d_{\vec{r}})] &= \sum_{f} \gamma(f) \sum_{(a,b) \in E} d_{\vec{r}}(f(a), f(b)) c(a,b) \\ &= \sum_{f} \gamma(f) \sum_{u,v \in K} d_{\vec{r}}(u,v) c_f(a,b) \\ &= \sum_{u,v \in K} d_{\vec{r}}(u,v) \sum_{f} \gamma(f) c_f(a,b) \\ &= \sum_{u,v \in K} d_{\vec{r}}(u,v) c'(u,v) \end{aligned}$$

So for all $d_{\vec{r}} \in D_G$, we have that $\sum_{u,v \in K} d_{\vec{r}}(u,v) c'(u,v) \leq \nu$ And using strong duality (and the dual for maximum concurrent flow given in Section 2):

$$cong_G(\vec{H}) = \sup_{d_{\vec{r}} \in D_G} \sum_{u,v \in K} d_{\vec{r}}(u,v) c'(u,v) \leq \nu$$

□

In fact, these flow sparsifiers are generated as a convex combination of 0-extension graphs G_f. As we noted earlier, this theorem can be immediately applied to the reduction in [22] to get an improved competitive ratio of Steiner oblivious routing:

Corollary 1. *There is an $O(\log^2 k/\log\log k)$-competitive Steiner oblivious routing scheme.*

Informally, this improvement comes from considering the question of finding an H that embeds with low-congestion into G directly, rather than constructing an H that approximates the terminal cut function and then relying on bounds on the max-flow min-cut ratio for maximum concurrent flow problems. This improves upon the previous bound by an $O(\log k)$ factor.

4 Improved Constructions

Here we give algorithms to construct $O(\log^2 k/\log\log k)$-quality flow sparsifiers in general graphs, and $O(\log k)$-quality flow sparsifiers for any graph that excludes a fixed minor. The quality of these flow sparsifiers improves upon the quality of the cut sparsifiers in previous results by a factor of $\tilde{O}(\log^{1.5} k)$ in both the general case and the case in which G excludes a fixed minor. We accomplish this by leveraging the improved existential result in Section 3 to write a stronger linear program for the problem of constructing a good quality flow sparsifier, and from this we can simplify the approximate separation oracles needed. The only remaining loss in quality compared to the existential result is the competitive ratio of the best oblivious routing scheme (on a graph of size k), which is $\Omega(\log k)$.

4.1 The Unit Congestion Polytope

As in Section 3.2, we can define the unit congestion polytope P_G with respect to G as:

$$P_G = \{\vec{f} \in \mathfrak{R}^{\binom{k}{2}} | cong_G(\vec{f}) \leq 1\}$$

We will also be interested in the boundary

$$S_G = \{\vec{f} \in \mathfrak{R}^{\binom{k}{2}} | cong_G(\vec{f}) = 1\}$$

Definition 9. *For all $\vec{f} \in \mathfrak{R}^{\binom{k}{2}}$ we will define $d_{\vec{f}}$ as a semi-metric that achieves*

$$\sum_{(u,v)\in E} d_{\vec{f}}(u,v)c(u,v) = 1 \text{ and } \sum_{(u,v)\in E} \vec{f}_{u,v} d_{\vec{f}}(u,v) = cong_G(\vec{f})$$

The existence of such a semi-metric is implied by strong duality. An α-quality flow sparsifier is a graph $H = (K, E_H)$ such that $P_G \subset P_H \subset \alpha P_G$.

We can interpret such a graph H as being a better congestion network than G, and yet not too much better. Formally, the inclusion formula above imply that any flow that is feasible in G must also be feasible in H, and any flow that is feasible in H must be feasible in G with congestion at most α.

4.2 Formulating Quality as a Linear Program

We can define a non-negative variable $x_{a,b}$ for each pair $a, b \in K$. We will write the constraints$P_G \subset P_H \subset \nu P_G$ as linear constraints on H. Consider the constraints $cong_H(\vec{r}) \leq cong_G(\vec{r}) = 1$ for all $\vec{r} \in S_G$. For any fixed $\vec{r} \in S_G$, define

$$D_{\vec{r}} = \{ \text{ all semi-metrics } d \text{ s.t. } \sum_{u,v} \vec{r}_{u,v} d(u,v) = 1\}$$

Suppose $cong_H(\vec{r}) > 1$. Then by strong duality, there is a $d \in D_{\vec{r}}$ such that $\sum_{u,v} c'(u,v)d(u,v) < 1$ so we can express the constraint $cong_H(\vec{r}) \leq cong_G(\vec{r}) = 1$ as

$$D = \cup_{\vec{r} \in S_G} D_{\vec{r}}$$

$$\sum_{u,v} c'(u,v)d(u,v) \geq 1 \text{ for all } d \in D$$

So given $\vec{r}$ such that $cong_G(\vec{r}) = 1$ and $cong_H(\vec{r}) > 1$ we can find a distance d such that

$$\sum_{u,v} c'(u,v)d(u,v) < 1 \text{ and } \sum_{u,v} \vec{r}_{u,v} d(u,v) = 1$$

by solving a linear program. Thus $d \in D_{\vec{r}} \subset D$, and given any $\vec{r}$ for which $cong_G(\vec{r}) = 1$ and $cong_H(\vec{r}) > 1$, we can find a violated constraint. Consider the constraint: $cong_G(\vec{r}) \leq \nu cong_H(\vec{r})$ for all $\vec{r} \in S_G$. We need to express this as a linear constraint on H too. This is equivalent to the question does H route in G with congestion ν? If it does not, then there is a semi-metric d on V s.t.

$$\sum_{(a,b) \in E} c(a,b)d(a,b) = 1 \text{ and } \sum_{u,v} c'(u,v)d(u,v) > \nu$$

So let R be the set of all semi-metrics so that $\sum_{(a,b)\in E} c(a,b)d(a,b) = 1$. The constraint $cong_G(\vec{r}) \leq \nu cong_H(\vec{r})$ can be expressed as $\sum_{u,v} c'(u,v)d(u,v) \leq \nu$ for all $d \in R$. And given H, if $cong_G(H) > \nu$ we can find a $d \in R$ (again by solving a linear program) such that $\sum_{u,v} c'(u,v)d(u,v) > \nu$ and we can locate a violated constraint.

4.3 Max-Min Congestion

We can check in polynomial time if $cong_G(H) > C \log k / \log\log k$, but how do we find a $\vec{r} \in S_G$ for which $cong_G(\vec{r}) = 1$ and $cong_H(\vec{r}) > 1$ if any such vector exists? This is exactly the Max-Min Congestion Problem:

Theorem 9. *[22] There is a polynomial time $O(\log k)$-approximation algorithm for the Max-Min Congestion Problem.*

So this implies:

Theorem 2. For any capacitated graph $G = (V, E)$ and any set $K \subset V$ of size k, there is a polynomial (in n and k) time algorithm to construct an $O(\log^2 k / \log\log k)$-quality flow sparsifier $H = (K, E_H)$. If G excludes a fixed minor, then there is a polynomial time algorithm to construct an $O(\log k)$-quality flow sparsifier.

5 Unit Congestion Polytopes of Expanders

We note that the result in Section 6 will subsume the result in this section, and readers interested only in the main lower bound in this paper can skip directly to Section 6. Here we present a simpler super-constant lower bound for the quality of flow-sparsifiers, but this lower bound requires an assumption on how a flow-sparsifier is generated. Yet this example will help us to illustrate a somewhat surprising geometric phenomenon associated with the unit congestion polytope of expander graphs. We exploit this phenomenon both here and in Section 6 to prove sub-optimal but super-constant congestion lower bounds against many multicommodity flows at once.

5.1 How to Prove Lower Bounds for Vertex Sparsifiers

As in Section 4.2, we associate each capacitated graph $H = (K, E_H)$ with a vector $\vec{x}$ so that the value of the non-negative variable $x_{a,b}$ is equal to the capacity of the edge connecting a and b in H. Then adopting the point of

view in Section 4.2, H is an α-quality flow-sparsifier for $G = (V, E)$, $K \subset V$ if and only if a particular set of linear inequalities on $x_{a,b}$ are satisfied. These linear inequalities come in two forms: constraints that every flow feasible in G is also feasible in H require that the capacities in the graph H not be too small, and constraints that every flow feasible in H is feasible in G with congestion at most α (which can be re-written as $cong_G(\vec{H}) \leq \alpha$) require that the capacities in H not be too large.

So let C_α be the set of linear constraints - i.e. $C_\alpha = (A_1, \vec{b}_1, A_2, \vec{b}_2)$, and H is an α-quality flow sparsifier if and only if the associated non-negative vector $\vec{x}$ satisfies $A_1\vec{x} \leq \vec{b}_1$ and $\vec{b}_2 \leq A_2\vec{x}$. Additionally, the entries in A_i, b_i are all non-negative. So G has an α-quality flow sparsifier if and only if the polytope $W = \{\vec{x} | \vec{x} \geq \vec{0}, A_1\vec{x} \leq \vec{b}_1 \text{ and } \vec{b}_2 \leq A_2\vec{x}\}$ is non-empty.

Given this discussion, the canonical way to prove that a graph $G = (V, E)$, $K \subset V$ has no α-quality flow-sparsifier is to prove that W is empty. Using Farkas' Lemma, we can re-state this condition as G has no α-quality flow-sparsifier if and only if there is a non-negative vector $\vec{y} = (\vec{y}_1, \vec{y}_2) \geq \vec{0}$ such that

$$\vec{y}_1^T A_1 - \vec{y}_2^T A_2 \geq \vec{0} \text{ and } \vec{y}_1^T \vec{b}_1 - \vec{y}_2^T \vec{b}_2 < 0$$

Let's examine the constraint $A_1\vec{x} \leq \vec{b}_1$ more closely: These constraints ensure that the capacities in H are not too large. So these constraints are associated with the condition that $cong_G(\vec{H}) \leq \alpha$. We can use Strong Duality to re-write the condition $cong_G(\vec{H}) \leq \alpha$ as: for all metrics $d : V \times V \to \Re^+$ such that $\sum_{(u,v)} c(u, v)d(u, v) \leq 1$, we have that $\sum_{(a,b)} c_H(a, b)d(a, b) \leq \alpha$. So in general, the system $A_1\vec{x} \leq \vec{b}_1$ has one constraint for every metric d for which $\sum_{(u,v)} c(u, v)d(u, v) \leq 1$. Yet as we will observe, in the case in which G is an uniform capacitated expander graph we can actually choose d to be the (re-scaled) natural shortest path metric on G where each edge is assigned a distance $\frac{1}{|E|}$. We will actually be able to remove all the constraints $A_1\vec{x} \leq \vec{b}_1$ except the constraint corresponding to the natural shortest path metric, and a single constraint for each terminal. We have reduced the size of the system of linear constraints from exponential in n (there is one constraint for every vertex of P_G, see Section 3.2) to a single constraint for the natural shortest path metric (plus a trivial constraint for each terminal). Yet what is surprising is that we have preserved enough structure in the system of inequalities that the system of inequalities is still infeasible for super-constant quality. So even restricting the scope of our proof to only consider the natural shortest path metric and a trivial constraint for each terminal (as opposed to all the constraints in $A_1\vec{x} \leq \vec{b}_1$), we are still able to derive a contradiction and prove that there is no constant quality flow-sparsifier.

We will refer to this single linear constraint corresponding to the natural shortest path metric as an *oblivious dual certificate*, since we use the constraint to prove lower bounds for $cong_G(\vec{H})$ yet the certificate is oblivious to $\vec{H}$. In the next subsection we will give a super-constant lower bound on the quality of the flow-sparsifier for any expander graph, provided that the flow-sparsifier is generated as a convex combination of 0-extension graphs G_f. This additional restriction on how H is generated corresponds to augmenting the system $A_2\vec{x} \leq \vec{b}_2$ so that not only do we enforce that $P_G \subset P_H$, but also that H is realized as a distribution on 0-extension graphs. As we will see in Section 7, this additional restriction implicitly requires H to not only be a better flow-network than G, but also to be a better network for network coding. So the restriction that H be generated as a convex combination of 0-extension graphs is a serious one, because there may be many other properties that H is (implicitly) required to preserve if it is required to be generated from contractions and not just required to be a better flow network.

Yet in Section 6, we will be able to lift the assumption that H be generated as a convex combination of 0-extension graphs. But the intuition that underlies both proofs is the same: For expander graphs we can ignore almost all the constraints in the system $A_1\vec{x} \leq \vec{b}_1$ and consider only the shortest path metric (and trivial constraints for each terminal). Yet ignoring the other constraints has not sacrificed too much already: the system will still be infeasible for super-constant quality. The abstract difference between the proof that we present in this section, and the one in Section 6 is that the way we derive a contradiction from this reduced system of inequalities is different. Here, the way we are able to derive a contradiction is more direct because we can use the assumption on H. Yet, when H can be generated arbitrarily, we will need to contend with many more pathologies before deriving a contradiction.

5.2 A Super-Constant Lower Bound with Assumption

Definition 10. *For a* 0*-extension* f *and* $a \in K$, $f^{-1}(a) = \{u \in V | f(u) = a\}$ *and size*$(a) = |f^{-1}(a)|$

So the size of a terminal $a \in K$ is just the number of vertices in V that are mapped to a by the 0-extension f.

We let $G = (V, E)$ be a constant degree expander on n nodes and choose *any* k terminals where $k = \frac{n}{\sqrt{\log^* n}}$. We need to prove that for every graph $G' = \sum_f \gamma(f) G_f$, the minimum congestion embedding of G' into G is super-constant. Strong duality implies that for any graph G', there is a dual certificate (i.e. a metric space) that provides a lower bound on the congestion of embedding G' into G, and this certificate actually achieves the optimum congestion. Optimal dual certificates for different flows can look quite different, and we can't hope to solve a linear program to find an optimal dual certificate for every possible G' in the space of our proof!

Instead, we give a single sub-optimal dual certificate that proves every $G' = \sum_f \gamma(f) G_f$ (that does not have too much total capacity) requires super-constant congestion (at least $\Omega(\sqrt{\log^* k})$) in order to embed into G. We refer to such a certificate, informally, as an oblivious dual certificate. Such a certificate provides a super-constant lower bound for the quality of many flow sparsifiers at once.

Lemma 2. *There is a fixed metric space* d *such that for all* 0*-extensions* f*, either* d *certifies that* G_f *requires congestion at least* $\Omega(\sqrt{\log^* k})$ *in order to embed in* G *or* G_f *has at least* $\frac{n}{10}$ *edges.*

Why does this lemma imply what we are after? In general, given two demands $\vec{f_1}$ and $\vec{f_2}$ (on just the terminal set), if we have a lower bound of C for the congestion of embedding each $\vec{f_i}$ into G this does not imply a lower bound of C for the congestion of embedding a convex combination of $\vec{f_1}$ and $\vec{f_2}$ into G. The function $cong_G()$ is sub-additive. But if the metric spaces that certify a lower bound of C for $\vec{f_1}$ and $\vec{f_2}$ respectively, are the same then this metric space also certifies a lower bound of C for any convex combination of $\vec{f_1}$ and $\vec{f_2}$. This is easy to see from the dual (given in Section 2x) to the maximum concurrent flow problem, because any metric space d certifies that the congestion of embedding $\vec{f}$ into G is at least

$$\frac{\sum_{a,b \in K} \vec{f^{a,b}} d(a,b)}{\sum_{(u,v) \in E} c(u,v) d(u,v)}$$

and this is a linear function in the demand vector $\vec{f}$.

In fact, the oblivious dual certificate that yields the above lemma is particularly simple: Choose d to be the shortest path metric (on G) resulting from placing a unit of distance on every edge in G.

Roughly, we will prove the above lemma by assuming that the metric space d does not certify that G_f requires congestion at least $\Omega(\sqrt{\log^* k})$ in order to embed in G. This will constrain how the neighborhood of any terminal in G_f can grow, and we will piece these local constraints together to demonstrate that an appropriately rooted breadth-first search tree cannot grow too quickly, and thus G_f contains too many edges.

Proof:

We prove this through contradiction. Suppose there is a 0-extension f for which there are at most $\frac{n}{10}$ total edges (counted with multiplicity) in G_f, and the cost of the oblivious dual certificate against G_f is at most $O(\sqrt{\log^* k})$. Given such a 0-extension f we will construct a breadth-first search tree in G_f. First, we decide how to root such a breadth first search tree. We first prove that there is a terminal $a \in K$ of size at least $\frac{3n}{4}$. This will be the root of our breadth-first search tree:

Claim 6. *If* G_f *has at most* $\frac{n}{10}$ *edges (counted with multiplicity), then there is a terminal* $a \in K$ *of size at least* $\frac{3n}{4}$

Proof: We prove this by contradiction: assume that there is no terminal a of size at least $\frac{3n}{4}$. We know that $\sum_a \text{size}(a) = n$ because any 0-extension f is a clustering of all $|V| = n$ nodes in G. And in particular, there is at most

one terminal a of size strictly greater than $\frac{n}{2}$. And the size of any such terminal is at most $\frac{3n}{4}$, so this implies that

$$\sum_{a \text{ s.t. } \text{size}(a) \leq \frac{n}{2}} \text{size}(a) \geq \frac{n}{4}$$

And G is an expander (that expands up to subsets of size $\frac{n}{2}$) so we know that each subset $f^{-1}(a)$ for a s.t. $\text{size}(a) \leq \frac{n}{2}$ has at least $\text{size}(a)$ incident edges. So if we count up the number of edges in G_f, there must then be at least $\frac{n}{8}$ edges (because by adding up all the incident edges to any set a of size at most $\frac{n}{2}$ we have at most double counted edges). This is a contradiction, because we have assumed that G_f has at most $\frac{n}{10}$ edges counted with multiplicity □

So there is a terminal a of size at least $\frac{3n}{4}$, and of course such a terminal is unique. So we construct the breadth-first search tree (starting at the root a) in the graph G_f. Let this tree be T_a. Each node in T_a is a terminal. We assume that the constraint

$$\frac{\sum_{(a,b)\in E_f} c_f(a,b)d(a,b)}{\sum_{(u,v)\in E} c(u,v)d(u,v)} = O(\sqrt{\log^* k})$$

is satisfied and from this we will give a recurrence for how quickly T_a can grow at each depth depending on the number of terminals at earlier depths.

Let E_f^i be the edges of T_a that join levels i and $i+1$. By assumption (and because $E_f^i \subset E_f$):

$$\frac{\sum_{(a,b)\in E_f^i} c_f(a,b)d(a,b)}{\sum_{(u,v)\in E} c(u,v)d(u,v)} \leq O(\sqrt{\log^* k})$$

And from this, we will prove that there are at least $L = O(\sqrt{\log^* k})$ levels in the breath-first search tree T_a before T_a reaches at least $\frac{k}{2}$ terminals.

Claim 7. *If all nodes in G have at most constant degree then for any node $u \in V$, and any natural number p, for any set $A \subset V$ of p distinct nodes (not containing u),*

$$\sum_{a\in A} d(u,a) \geq \Omega(p \log p)$$

Proof: Given any node $u \in V$, given any set A of p distinct nodes (not containing u), we can minimize the total distance to u by greedily choosing any node that is closest to u among all nodes not yet selected. But the set of nodes at distance $O(\log p)$ from u is at most $\frac{p}{2}$, so this implies the claim. □

We can immediately apply this claim, to show that T_a cannot grow too quickly, otherwise there will be some level i where just the edges connecting level i to level $i+1$ provide a large value for the oblivious dual certificate already. So suppose that there are r terminals on level i of T_a. For each terminal i_q in this set $\{i_1, i_2, ...i_r\} \subset K$ let d_{i_q} be the number of distinct terminals in level $i+1$ that i_q is adjacent to (in G_f). Using the above claim:

$$\sum_{(a,b)\in E_f^i} c_f(a,b)d(a,b) \geq \Omega(\sum_{i_q} d_{i_q} \log d_{i_q})$$

So because G_f does not have too large a cost against the oblivious dual certificate (and $\sum_{(u,v)\in E} c(u,v)d(u,v) = \Theta(n)$) this implies that $\sum_{i_q} d_{i_q} \log d_{i_q} = O(n\sqrt{\log^* k}) = O(k \log^* k)$. If we fix $\sum_{i_1} d_{i_q} = D_i$, then $\sum_{i_q} d_{i_q} \log d_{i_q}$ is minimized for $d_{i_q} = \frac{D_i}{r}$. So this implies that

$$D_i \log \frac{D_i}{r} = O(k \log^* k)$$

Note that the number of terminals at level $i+1$ is at most D_i. Let $m_1, m_2, ...$ be the number of terminals at level $1, 2, ...$ respectively. Then we get the constraint $m_{i+1} \log \frac{m_{i+1}}{m_i} = O(k \log^* k)$. This constraint implies that the number of terminals at a given level is maximized by making the number of terminals on every earlier level as large as possible subject to the constraint. We can compute the recurrence:

$$m_1 = 1, m_2 = O(k\frac{\log^* k}{\log k}), m_3 = O(k\frac{\log^* k}{\log\log k}),$$

$$m_4 = O(k\frac{\log^* k}{\log\log\log k}), ... \; m_i = O(k\frac{\log^* k}{\log^{(i)} k})$$

where $\log^{(i)} k$ is the iterated logarithm. So in order to reach at least $\frac{k}{2}$ nodes, $L = \Omega(\log^* k)$.

But if after $L = \Omega(\log^* k)$ levels, the breadth-first search tree T_a has reached at most $\frac{k}{2}$ terminals, then we can give a lower bound on the number of edges in G_f using the expansion properties of G: Consider level i, and let $A_i \subset K$ be the set of terminals that have not yet been reached by the breadth-first search tree in the first i levels. So $|A_i| \geq \frac{k}{2}$ for all $i \leq L$. So this implies that $size(A_i) \geq \frac{k}{2}$, where the size of a subset of terminals is defined to be the sum of the sizes of each terminal in the set. Because $a \notin A_i$, and the size of a is at least $\frac{3n}{4}$, this implies that $size(A_i) \leq \frac{n}{4}$. So the set $f^{-1}(A_i)$ expands, and is incident to at least $size(A_i) \geq \frac{k}{2}$ edges. Because the tree T_a is a breadth-first search tree, each edge with exactly one endpoint in $f^{-1}(A_i)$ must join level i to level $i+1$ in the breadth-first search tree. So if we sum the number of edges with exactly one endpoint in $f^{-1}(A_i)$ over all $i \leq L$, we have counted each edge in G_f at most once, and we have counted at least $\frac{kL}{2} = \omega(n)$ edges. And this is not possible, because there are at most $\Theta(n)$ edges in G_f.

□

We can leverage this lemma to prove an $\Omega(\sqrt{\log^* k})$ lower bound on the quality of flow sparsifiers for G, provided that these flow sparsifiers are generated as a convex combination of 0-extension graphs, but this is not the result we are after.

Theorem 10. *For any distribution on 0-extensions γ, the congestion of embedding $G' = \sum_f \gamma(f) G_f$ into G is at least $\Omega(\sqrt{\log^* k})$*

Proof: Consider a distribution γ on 0-extensions. Let $G' = \sum_f \gamma(f) G_f$, and by applying Markov's Inequality either G' has at most $\frac{n}{20}$ total units of capacity, or for at least $\frac{1}{2}$ probability γ results in a 0-extension f for which G_f has at most $\frac{n}{10}$ edges.

Consider the latter case: We can construct a lower bound for the congestion of embedding G' into G using the dual certificate d and weak duality. The congestion is at least

$$\frac{\sum_{(a,b)\in E'} c'(a,b) d(a,b)}{\sum_{(u,v)\in E} c(u,v) d(u,v)} = \sum_f \gamma(f) \frac{\sum_{(a,b)\in E_f} c_f(a,b) d(a,b)}{\sum_{(u,v)\in E} c(u,v) d(u,v)}$$

And

$$\sum_f \gamma(f) \frac{\sum_{(a,b)\in E_f} c_f(a,b) d(a,b)}{\sum_{(u,v)\in E} c(u,v) d(u,v)} \geq Pr_{f\leftarrow\gamma}[G_f \text{ has at most } \frac{n}{10} \text{ edges }] \Omega(\sqrt{\log^* k}) = \Omega(\sqrt{\log^* k})$$

And this yields the desired lower bound. Consider the former case: there must be some terminal $a \in K$ incident to at least $\frac{n}{10k}$ units of capacity in G'. But the minimum cut in G separating a from $K - \{a\}$ is constant, so the total demand in $\vec{G'}$ incident to terminal a is at least $\Omega(\frac{n}{k}) = \Omega(\sqrt{\log^* k})$ but the min-cut separating a from $K - \{a\}$ is constant so the congestion of embedding G' into G is at least $\Omega(\frac{n}{k}) = \Omega(\sqrt{\log^* k})$. □

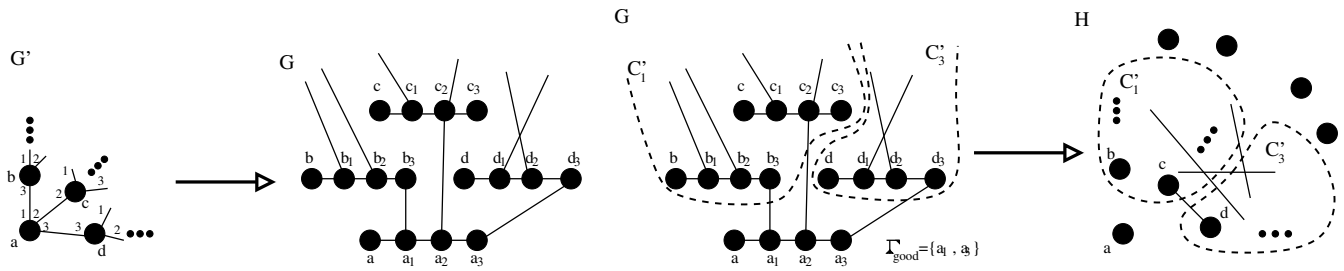

Figure 1. a. Constructing the graph $G = (V,E)$ from an x-regular, high-girth expander $G' = (K, E(G'))$
b. Constructing the partition $\{C'_i\}$ of terminals at distance at most D

The proof of our main theorem will follow the same pattern as outlined above: Instead of considering the full system of inequalities that corresponds to the constraint $cong_G(\vec{H}) \leq \alpha$, we will choose a single inequality corresponding to the natural shortest path metric (and again, the underlying graph is an expander). This linear inequality will constrain how the neighborhood of a terminal in a good flow sparsifier can grow. We can then piece together these local constraints to get a global contradiction. Of course here we were able to use a natural notion of how quickly the neighborhood of a terminal grows, because we computed the breadth-first search tree and interpreted the oblivious dual certificate as a constraint on each level in the tree. Yet when we place no restriction on how a flow sparsifier is generated, we will require a more intricate notion of "how a neighborhood grows" that will revolve around how much flow can be sent to neighbors on only short paths, using local paths.

6 Limits to Flow Sparsification

Here we give a graph $G = (V,E)$ and a subset $K \subset V$ of size k for which every flow sparsifier H has quality $\Omega(\log\log k)$. This lower bound is unrestricted, and makes *no* assumptions on how H is generated.

6.1 The Graph G

Here we define the graph $G = (V,E)$, and highlight the properties of G that will be used in proving an $\Omega(\log\log k)$ lower bound using this graph. G is generated from another graph $G' = (K, E(G'))$. Set $x = \log^{1/3} n$ and let $k = \frac{n}{x}$. We choose $G' = (K, E(G'))$ to be an x-regular graph that has

$$girth(G') \geq \Omega(\frac{\log k}{\log x}) = \Omega(\frac{\log n}{\log\log n})$$

For each node $u \in G'$, we denote the set of neighbors of u in G' as $\Gamma(u)$. Additionally, order this set arbitrarily (for each node u), and let $\Gamma^i(u)$ be the i^{th} neighbor of u in G'. This labeling scheme need not be consistent, and node u can label a neighbor v as the i^{th} neighbor ($v = \Gamma^i(u)$) but v labels u as the j^{th} neighbor ($u = \Gamma^j(v)$).

We construct $G = (V,E)$ from G': For each terminal $u \in G'$, construct a path $P_u = u, u_1, u_2, ...u_x$ in G. And for each edge $(u,v) \in G'$, suppose that $v = \Gamma^i(u), u = \Gamma^j(v)$, add an edge in G connecting u_i and v_j. See Figure 1*a*.

Claim 8. *Any matching M in G' can be routed in G - i.e. $cong_G(\vec{M}) \leq 1$*

Definition 11. *Let C_G be the total capacity in G and let $\Delta(G)$ be the maximum degree of G. Also let C_H be the total capacity in H.*

Definition 12. *Let $dist_G(u,v)$ and $dist_{G'}(u,v)$ be the natural shortest path distances on G, G' respectively*

We will need a number of properties of G in order to prove an $\Omega(\log\log k)$ lower bound:

1. $C_G \leq 2n$
2. $\Delta(G) = 3$
3. $\forall_{u,v\in K} dist_G(u,v) \geq dist_{G'}(u,v)$
4. $girth(G) \geq girth(G')$

We will first give some observations that are the foundation of the proof: Suppose there is a flow sparsifier H that has $O(1)$-quality. This implies that $cong_G(\vec{H}) = O(1)$, and we can enumerate some properties of H. These are the properties that we will use to derive a contradiction (and in fact get a quantitative bound $cong_G(\vec{H}) \geq \Omega(\log\log k)$):

1. The total capacity in H must be $O(k)$

   If H had $\omega(k)$ total capacity, then there would be some terminal $a \in K$ that is incident to $\omega(1)$ total capacity in H. Yet the minimum cut in G separating a from $K - \{a\}$ is constant (G is a constant degree graph), so this would immediately yield a super-constant lower bound on the congestion of embedding H into G.

2. Most capacity in H must be between terminals for which the distance is much smaller than the girth of G

   Suppose, for example, that H is a complete graph with uniform edge capacities of $\frac{1}{k}$. Then the average distance (as measured in G) between two random terminals is roughly $\Omega(girth(G'))$. So if we choose the natural shortest path distance in G as the oblivious dual certificate, then

   $$\sum_{a,b\in K} c_H(a,b)d(a,b) \geq \Omega(girth(G')k)$$

   Yet $\sum_{(u,v)\in E} c(u,v)d(u,v) = \Theta(n)$ and this would yield a super-constant lower bound on the congestion of embedding H into G.

3. Fix the optimal embedding of H into G. Then at most $o(k)$ of the total capacity in H can be routed (in the optimal embedding) along high-girth paths.

   Suppose, for example, that $\Omega(k)$ total capacity in H is routed in G using high-girth paths. The total capacity in G is $\Theta(n)$, yet each unit of capacity in H that is routed along a high-girth path uses up $\Omega(girth(G))$ total capacity in G and $\Omega(girth(G)k)$ total capacity is consumed, but this is $\omega(n)$. So in fact, for most edges (a,b) in H, most of the $c_H(a,b)$ flow in the optimal embedding must traverse a path much shorter than the girth of G, so this flow must traverse the natural shortest path between a and b.

Additionally, the requirement that H can route any matching in G' places constraints on the local neighborhood of a terminal.

A There are $\omega(1)$ neighbors of a in G', and a can route a unit flow to each such neighbor using edges in the graph H

   H cannot hope to support such flows via a direct edge, because this would require a to be incident to $\omega(1)$ total capacity in H. So the paths that support these flows must be indirect.

B In some sense, on average the path used to route a unit flow from a to b in H must be constant length (using an argument similar to Condition 2 above)

   If, on average, the length of any such path were $\omega(1)$, then if when we route a matching in G' in H, there are $\Theta(k)$ total edges in the matching and $\Theta(k)$ total units of flow. But each unit of flow would use up $\omega(1)$ total capacity in H, because the paths on average are $\omega(1)$ length in H.

C So a must be able to send a unit flow to a super-constant number of terminals, using constant length paths. Additionally these paths must not reach any terminal b that is too far from a in G.

So the local neighborhood of a must be able to support all these requirements - a must be able to send a unit of flow to a super-constant number of terminals, using constant length paths that remain within the local neighborhood. Yet we know how most of the edges in the local neighborhood of H embed (in the optimal embedding) into G, because these edges mostly use the natural shortest path in G. This shortest path traverses a sub-interval of P_a. So we can derive a contradiction, because any local neighborhood that can support all these requirements must have large cut-width, and so the natural embedding of this neighborhood would incur super-constant congestion on some sub-interval of P_a.

Roughly, the outline of the argument is the same as in the previous section: we use oblivious dual certificates to establish local constraints on how the neighborhood of any terminal a. We can piece these local constraints together (using the fact that all matchings in G' can be routed in H) to reduce this question to a more structured question about cut-width, and from this we derive a contradiction (Section 6.4).

6.2 Girth and Capacity Surgery

Let $H = (K, E(H))$ be an arbitrary flow sparsifier. We will argue that if all matchings M in G' can be routed in H with congestion at most 1, then $cong_G(\vec{H}) = \Omega(\log\log k)$. We do not assume anything structurally about H, so there are many different ways in which the demand $\vec{H}$ can result in super-constant congestion. We will need a complex argument to handle this, and ultimately we will reduce computing a lower bound on $cong_G(\vec{H})$ to a more structured embedding question about cut-width.

Assume $c_1 > 32 \times c_2$ and $c_2 > 24$

Given H, we fix the optimal embedding of H into G. This is a min-congestion routing of $\vec{H}$ in G. Each edge $(u,v) \in E(H)$ is mapped to a flow in G from u to v of value $c_H(u,v)$. We can decompose such a flow from u to v in G into paths. We refer to such a decomposition as a path decomposition, and we assume that this decomposition uses only simple paths.

Definition 13. *We call an edge $(u,v) \in E(H)$ girth-routed if at least $\frac{c_H(u,v)}{2}$ flow (of the $c_H(u,v)$ total flow) from u to v is routed using paths (in G) of length at least $\frac{girth(G')}{4}$.*

Definition 14. *Let $H' = (K, E(H'))$ be the graph defined by deleting all girth-routed edges in H. Let C_{gr} be the total capacity of girth-routed edges in H - i.e. the total capacity deleted from H to obtain H'.*

If H $O(1)$-approximates the congestion of all multicommodity flows in G, then the total capacity in H must be $O(k)$. This is true because each node $u \in K$ has degree 1 in G, and so if the total capacity in H is $\omega(k)$ then there is a node $u \in K$ which is incident to $\omega(1)$ total units of capacity, and even just routing the demands in $\vec{H}$ incident to u would incur $\omega(1)$ congestion in G.

Given this, we know that if H $O(1)$-approximates the congestion of all multicommodity flows in G, then on average edges (u,v) in G' must be able to route a unit flow from u to v using constant length paths. Otherwise we could construct a matching M in which all edges need super-constant length paths to route a unit flow in H, and this flow would not be feasible in H because there would be $\Omega(k)$ pairs each using $\omega(1)$ units of capacity - but H has only $O(k)$ total capacity.

We need to formalize this notion, that edges in G' must on average use short paths (and in fact a stronger condition, that edges in G' must also not route flow in H that reaches nodes that are too far away according to the shortest path metric in G'). So we set $D = \log^{2/3} n$.

Definition 15. *We call a path P connecting u and v (in H or H') good if P is length $< \frac{1}{c_2}\log\log n$ and reaches no terminal a for which either $dist_{G'}(u,a)$ or $dist_{G'}(v,a) \geq D$.*

We call an edge $(u,v) \in E(G')$ good if u can send at least $\frac{1}{2}$ unit of flow to v using only good paths in H', and otherwise we call edge (u,v) bad.

Note that the definition of a good edge is with respect to flows in H', not H. We need this for a technical reason.

Case 1

Suppose that $C_H \geq \frac{k}{c_1} \log\log n$

Then there is a node $u \in K$ that has at least $\frac{2}{c_1} \log\log n$ total capacity incident in H. And because v is degree one in G, this implies that $cong_G(\vec{H}) \geq \frac{2}{c_1} \log\log n$ so we can conclude that $C_H \leq \frac{k}{c_1} \log\log n$.

Case 2

Suppose that $C_{gr} \geq \frac{k}{\log^{1/3} n}$.

Then, in the optimal embedding of H into G there is at least $\frac{C_{gr}}{2} \geq \frac{k}{2\log^{1/3} n}$ total capacity that uses paths of length at least $\frac{girth(G')}{4} = \Omega(\log n / \log\log n)$. So the total capacity used in routing H into G is $\Omega(n \log^{1/3} n / \log\log n)$. And $C_G \leq 2n$ so this implies:

$$cong_G(\vec{H}) \geq \Omega(\frac{\log^{1/3} n}{\log\log n})$$

So we can conclude that $C_{gr} \leq \frac{k}{\log^{1/3} n}$

6.3 Matchings that are Difficult to Route

We will use the following elementary lemma to simplify the case analysis:

Lemma 3. *Let G' be an x-regular graph on k terminals. Let $F \subset E(G')$, s.t. $|F| \geq ckx$. Then there is a matching $M \subset F$ s.t. $|M| \geq \frac{c}{2}k$*

Proof: Let M be a maximal matching using only edges in F. Then let A be the set of terminals that are matched in M, and let B be the remaining terminals. No edge in F can have both endpoints in B, because otherwise M would not be maximal. So all edges in F are adjacent to a terminal in A. Because G' is x-regular, the number of edges in $E(G')$ adjacent to some terminal in A is at most $|A|x$, and this must be at least the size of F. So $|A|x \geq ckx$ In particular, $|A| \geq ck$, and every terminal in A is matched in M so there are at least $\frac{c}{2}k$ edges in M. □

Case 3

Suppose that there are at least $\frac{kx}{4}$ bad edges (in G'). We let F be the set of bad edges in G'. $|F| \geq \frac{kx}{4}$. We can use the above lemma to find a matching $M \subset F$ so that $|M| = \frac{k}{8}$. M is a matching, so $\vec{M}$ is routable with congestion at most 1 in G, and so must also be routable with congestion at most 1 in H.

For each edge $(u,v) \in M$, set $y_{u,v} = 1$. Then using Case 2, $C_{gr} \leq \frac{k}{\log^{1/3} n} = o(k)$. Deleting any edge $(a,b) \in E(H)$ of capacity $c_H(a,b)$ affects at most $c_H(a,b)$ units of flow in the routing of $\vec{M}$ in H because $\vec{M}$ can be routed in H with congestion at most 1. So we can set $y'_{u,v}$ as the amount of flow remaining using the same routing scheme for $\vec{M}$ in H' as for $\vec{M}$ in H, just deleting paths that traverse some girth-routed edge in H (which is deleted in order to obtain H').

$|M| = \frac{k}{8}$ and so $\sum_{u,v} y_{u,v} = \frac{k}{8}$. Because $C_{gr} \leq o(k)$, $\sum_{u,v} y'_{u,v} = \frac{k}{8} - o(k)$. Let $\vec{f}'(M)$ be the flow in H' that satisfies $y'_{u,v}$ that results from deleting all flow paths that traversed a girth-routed edge in H. Each edge $(u,v) \in M$ is bad, so if we decompose $\vec{f}'(M)$ into paths, then for any $(u,v) \in M$ at most $\frac{1}{2}$ unit of flow is carried by good paths. So the total flow carried by bad paths is at least

$$\sum_{u,v} y'_{u,v} - \frac{1}{2}|M| \geq \frac{k}{8} - o(k) - \frac{1}{2}|M| = \frac{k}{16} - o(k)$$

So there must be $\frac{k}{16}$ total units of flow on bad paths in H', and in particular there must be at least $\frac{k}{32}$ total units of flow on either paths of length at least $\frac{1}{c_2}\log\log n$ or at least $\frac{k}{32}$ total units of flow on paths that reach nodes a that are at least distance D from one of the two endpoints u or v of the flow. We consider these two sub-cases separately:

Case 3**a**

Suppose that there are $\frac{k}{32}$ total units of flow on paths in H' of length at least $\frac{1}{c_2}\log\log n$. Notice that the total capacity in H', $C_{H'}$ is at most the total capacity in H, C_H (because H' resulted from H by deleting girth-routed edges). And the total capacity in H' is at least $\frac{k}{32\times c_2}\log\log n$. But then

$$C_{H'} \geq \frac{k}{32\times c_2}\log\log n > \frac{k}{c_1}\log\log n \geq C_H$$

using Case 1, and we have a contradiction

Case 3**b**

Suppose that there are $\frac{k}{32}$ total units of flow on paths in H' that reach a node a that is distance at least D from one of the two endpoints u or v of the path.

Consider any such path, $P_{u,v}$ in H' that is carrying λ units of flow from $y'_{u,v}$. Suppose that a is the first node such that $dist_{G'}(u,a) \geq D$ or $dist_{G'}(v,a) \geq D$. Assume without loss of generality that $dist_{G'}(u,a) \geq D$. Construct a demand vector $\vec{r} \in \mathfrak{R}^{\binom{k}{2}}$ as follows: For each such path $P_{u,v}$, increment the demand $\vec{r}_{u,a}$ by λ.

So for each flow path carrying λ units of flow, we have added λ demand to some coordinate in $\vec{r}$.

Claim 9. *$\vec{r}$ is routable in H with congestion at most* 1

Proof: We constructed $\vec{r}$ demands based on a flow $\vec{f'}(M)$ of congestion at most 1 in H', and the same routing scheme for $\vec{f'}(M)$ that achieved congestion at most 1 in H' also achieves congestion at most 1 in H. We then constructed demands by deleting segments of the flow paths - i.e. in the above example we preserved only the sub-path of $P_{u,v}$ from u to a and deleted the rest of the path. This flow resulting from deleting sub-paths still has congestion 1 in H and also satisfies the demands $\vec{r}$. □

Notice that for every path carrying λ flow we have added a demand equal to this flow value λ between two nodes $u, a \in K$ which are distance at least D apart in G' - i.e. $dist_{G'}(u,a) \geq D$.

Consider the minimum congestion embedding of the demand vector $\vec{r}$ into G. We can use the natural shortest path metric on G to certify $\vec{r}$ has large congestion in G: for each edge in G, place a unit of distance on this edge.

The total distance × capacity units used is $C_G \leq 2n$. And we note that $dist_G(u,v) \geq dist_{G'}(u,v)$ for all $u, v \in K$ from the claim Section 6.1. So $\sum_{u,v} \vec{r}_{u,v} dist_G(u,v) \geq \frac{k}{32}D$.

This implies $cong_G(\vec{H}) \geq cong_G(\vec{r}) \geq \frac{\Omega(k)D}{2n} = \Omega(\log^{1/3} n)$. So we can conclude using Case 3a and 3b that there are at most $\frac{kx}{4}$ bad edges in G'.

6.4 Cut-Width

So we can remove all bad edges in G' from consideration, and there are at least $\frac{kx}{4}$ good edges remaining. This implies that there is a terminal $a \in G'$ that is incident to at least $\frac{x}{2}$ good edges. Consider such a terminal a.

Let $\Gamma_{good}(a)$ be the set of neighbors u of a for which the edge $(u,a) \in E(G')$ is good. We can define the induced path P_{good} as the path on just Γ_{good} in the order in which these nodes are labeled according to Γ. For each terminal b adjacent to a in G', we define a component C_j (assuming $b = \Gamma^j(a)$) that consists of all terminals that are distance at most D according to G' from terminal a and are in the depth-first search sub-tree (of G') rooted at terminal b. For each terminal b adjacent to a s.t. the edge (a,b) is bad, if $b = \Gamma^j(a)$ then choose the closest terminal c on the path P_a for which the edge (c,a) is good, and break ties arbitrarily. For each node $c \in \Gamma_{good}$, consider the set of

terminals d that chose c, and set $C'_i = \cup_{d \text{ chooses } {}_cC_j}$ if $\Gamma^i(u) = c$ and $\Gamma^j(u) = d$, and by definition any good terminal on the path P_a chooses itself.

So C'_i forms a partition of the set of terminals that are distance at most D (since $girth(G') >> D$) from terminal a in G', and for each component C'_i there is a terminal $c \in C'_i$ that is adjacent to a in G' and for which (a,c) is a good edge.

We construct a graph B in which we add a node for a, and a node i for each component C'_i. For each edge in H' from a node C'_i to a node in C'_j, we add an edge in B of the same capacity connecting the node i to the node j (and we allow parallel edges). See Figure 1*b*.

Intuitively, in B we allow routing within C'_i for free. But because each node i in B contains a terminal c which is a neighbor of a and for which (a,c) is a good edge, terminal a must be able to send at least $\frac{1}{2}$ units of flow from a to terminal c in H' using paths that do not reach any other node distance more than D from terminal a (according to $dist_{G'}$) and using only paths of length at most $\frac{1}{c_2}\log\log n$. Because in B we allow routing within C'_i for free, and have only removed nodes that are too far from a to be on any good path, in the graph B we must be able to send at least $\frac{1}{2}$ units of flow of flow from a to i using paths of length at most $\frac{1}{c_2}\log\log n$.

So consider the path P_{good} defined on $\{a\} \cup \Gamma_{good}(a)$ in which nodes in Γ_{good} are visited according to the order defined by $\Gamma(a)$. We will prove a lower bound on the minimum cut-width embedding of B into P_{good}. This will imply a cut-width lower bound on embedding B into P_a, and because every edge in H' routes at least $\frac{1}{2}$ of its capacity through paths of length at most $\frac{girth(G')}{4}$ (and consequently every edge in B connecting a node i to a node j must traverse the sub-path $a_i, a_{i+1}, ...a_j$ in P_a), and so this implies a lower bound on the congestion of embedding H' into G. So consider an interval I of P_{good}:

Definition 16. *Given a set of flows out of a, we define the average path length as the sum over all flow paths P (in a path decomposition of the flow) of the weight of flow on the path times the length of the path: $v(P) \times length(P)$.*

Note that the average path length of a flow is independent of the particular choice of the path decomposition of the flow, provided every path in the path decomposition is simple.

Definition 17. *We define the minimum cost-routing into an interval I as the minimum average path length of any flow (in B) that sends exactly $\frac{1}{2}$ unit of flow in total to the terminals in interval I.*

Definition 18. *An interval I has the (C,E) property if the total capacity (of B) crossing I - i.e. has one endpoint to the left and one endpoint to the right of I - is at least C and if the minimum cost routing to I is at least E.*

Suppose that interval I has the (C,E)-property. Split I into I_1, I_2, I_3, three contiguous intervals. If there is total capacity at least $\frac{1}{6}$ in B crossing I_1 but not I, then I_1 has the $(C + \frac{1}{6}, E)$ property (because the minimum cost routing to I_1 is at least the minimum cost routing to I). Similarly, if there is at least $\frac{1}{6}$ capacity in B crossing I_3 but not I, then I_3 has the $(C + \frac{1}{6}, E)$ property.

Otherwise, there is at most $\frac{1}{6}$ total capacity entering I_2 from the left of I_1, and at most $\frac{1}{6}$ total capacity entering I_2 from the right of I_3. So consider the minimum cost routing to I_2. There must be at least $\frac{1}{2} - \frac{1}{6} - \frac{1}{6} = \frac{1}{6}$ total flow that enters I_2 from I_1 or I_3. So given the minimum cost routing to I_2 of cost α, we can stop paths that enter I_2 from I_1 or I_3 short, and reduce the cost of the routing by at least $\frac{1}{6}$. This would give a routing into I of cost $\alpha - \frac{1}{6}$, but the minimum cost routing into I is at least E, so $\alpha \geq E + \frac{1}{6}$. So either I_1 or I_3 has the $(C + \frac{1}{6}, E)$ property or I_2 has the $(C, E + \frac{1}{6})$ property.

So this implies that if we continue for $\log_3 \frac{x}{2}$ levels of this recursion (and there are in fact $\frac{x}{2}$ nodes in the path P_{good}), then we find an interval I that has the (C,E) property and $C + E \geq \frac{\log_3 \frac{x}{2}}{6}$ and so either C or E must be $\geq \frac{\log_3 \frac{x}{2}}{12}$.

If $C \geq \frac{\log_3 \frac{x}{2}}{12}$, then the cut-width of embedding B into the path P_{good} is at least C, which implies that at least C total capacity in H' has one endpoint in C'_i and one endpoint in C'_j, and the interval I in P_a is contained between

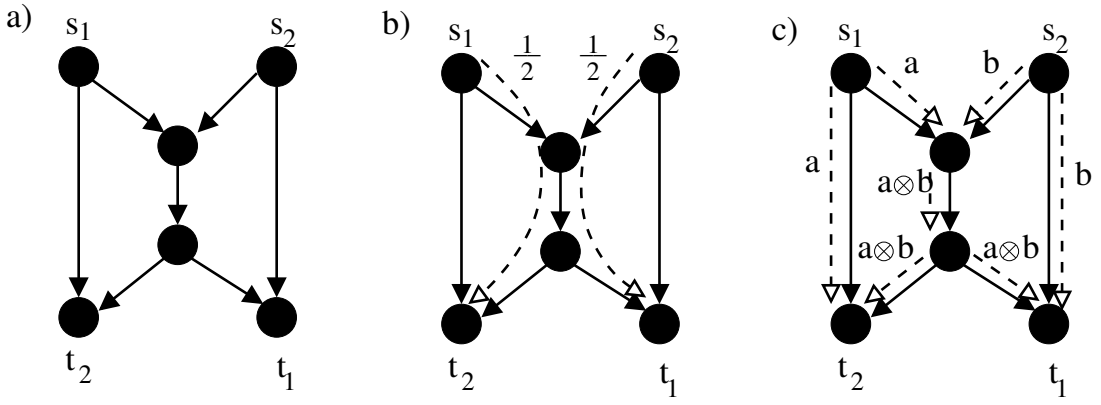

Figure 2. a. A communication problem on a directed graph b. A multicommodity solution c. A network coding solution

the image of C'_i in P_a and the image of C'_j in P_a. Because this capacity is in H', and at least $\frac{1}{2}$ of this capacity is routing using the natural path in G and traverses the interval I, and in this case the congestion of embedding H' into G (using the fixed routing of H' into G - note that H' was defined based on the optimal embedding of H into G) is at least $\frac{C}{2} = \Omega(\log \log n)$.

If $E \geq \frac{\log_3 \frac{x}{2}}{12}$, then there is a node $u \in I$ for which the minimum cost routing to c is at least $\frac{\log_3 \frac{x}{2}}{12} > \frac{\log \log n}{c_2}$, but then the edge (a,u) would not be good. I is an interval in P_{good} and this yields a contradiction.

Theorem 3. For infinitely many values of k, there is a graph $G = (V,E)$ and a set $K \subset V$ of size k for which any flow sparsifier has quality at least $\Omega(\log \log k)$

7 Network Coding

Suppose we are given the communication problem in Figure 2*a* - s_1 wants to send a message to t_1 and s_2 wants to send a message to t_2. This example is due to Ahlswede et al [3]. We can phrase this communication problem as a multicommodity problem - i.e as a maximum concurrent flow problem, in which case the optimal rate is $\frac{1}{2}$ as in Figure 2*b*. However, if we allow messages to be XORed, then the optimal rate can be made to be 1 as in Figure 2*c*. This later solution to the communication problem is a network coding solution.

We do not define the precise notion of a network coding problem or solution here because we will not use these definitions apart from noticing that flow sparsifiers that are generated as a convex combination of 0-extension graphs not only (approximately) preserve the multicommodity flow rate but also (approximately) preserve the network coding rate. See [15] for a precise definition of network coding rate.

Definition 19. *Given a demand vector $\vec{f} \in \Re^{\binom{k}{2}}$ (as in the case of a maximum concurrent flow problem), suppose that R is the network coding rate for the network coding problem defined by $\vec{f}$ and G (as in [15]) then we define the network congestion as $ncong_G(\vec{f}) = \frac{1}{R}$*

It is always true that $ncong_G(\vec{f}) \leq cong_G(\vec{f})$ because from any maximum concurrent flow solution we can construct a network coding solution that is at least as good. Also, from the above example we can have a strict inequality in the case of a directed graph - i.e. the network coding solution performs strictly better. However, it is conjectured [1], [15], [21] that the network coding rate is always equal to the maximum concurrent flow rate

for undirected graphs. Despite much work on this problem, researchers have only been able to improve upon the sparsest cut upper bound for the network coding rate in specific cases [1], [15], [16], [20].

This conjecture has far-reaching implications. In particular using [1] even proving an $o(\log k)$ bound - i.e. $cong_G(\vec{f}) \leq o(\log k) ncong_G(\vec{f})$ for all undirected graphs would imply the first super-linear lower bound for oblivious matrix transposition in the bit-probe model, and this would be the first super-linear lower bound in this powerful model for *any* problem.

We make some observations relating network coding to the problem considered in this paper - namely that of proving a lower bound for flow sparsifiers. Let $H = \sum_f \gamma(f) G_f$ be a flow sparsifier generated from a convex combination of 0-extension graphs.

Claim 10. *For all $\vec{f}$, $ncong_H(\vec{f}) \leq ncong_G(\vec{f})$*

Briefly, this is true because using any reasonable definition of network coding rate, performing contractions does not make any network coding solution infeasible. But again see [15] for a precise definition, from which the claim is self-evident.

Claim 11. *If $cong_G(\vec{H}) = C$, then for all $\vec{f}$, $ncong_G(\vec{f}) \leq C ncong_H(\vec{f})$*

The proof of this is almost identical to the proof of Claim 1, and just relies on using the embedding of H into G: This embedding can be composed with any network coding solution in H (say, of network congestion at most 1) to get a network coding solution in G in which the network congestion is at most C.

So the point of this discussion is that if we only consider flow sparsifiers generated from a convex combination of 0-extension graphs, then we also preserve the network coding rate in addition to the multicommodity flow rate.

Theorem 4. Any flow sparsifier generated as a convex combination of 0-extensions that has quality ν also preserves the rates of all network coding problems with endpoints supported in K to within a ν-factor.

So we can consider three different problems: that of generating a good quality flow sparsifier from a convex combination of 0-extension graphs (P1), generating a good quality flow sparsifier that preserves the network coding rate and the multicommodity flow rate (i.e. the quality is measured over all network coding problems and all multicommodity flow problems) (P2), and that of generating a good quality sparsifier that preserves only the multicommodity flow rate (P3).

The above Claims imply that the ordering of problems arranged in terms of decreasing difficulty is P1 $\geq$ P2 $\geq$ P3. If we wanted only to prove a (super constant) lower bound for the quality of flow sparsifiers generated from convex combinations of 0-extensions, the proof of a lower bound would be much more straight-forward (in fact we provide such a weaker proof in Section 6).

Yet the general reduction technique in [22] used to reduce a multicommodity-type problem to a problem on a graph of size k only requires that a flow sparsifier approximately preserve the congestion of all multicommodity flows. So really, the right question to ask (i.e. the question that has implications for approximation algorithms) is how well flow sparsifiers can approximate the congestion of all multicommodity flows, not how well flow sparsifiers can approximate multicommodity flows and network coding rates.

But any claim that P3 is as difficult as P1 would imply, at the very least, that the network coding rate is always at least as easy to preserve as the multicommodity flow rate. Nothing about the network coding rate in general undirected graphs (except that it is bounded by the sparsest cut) is known, and any convex combination of 0-extensions may also preserve more graph theoretic quantities other than just the network coding rate (as in the above Claims) and we don't even know what these quantities are, let alone anything interesting about them.

Roughly, any statement that P1 is equivalent to P3 would imply that any rate that is monotone increasing under contractions and closed under composition with flow-like embeddings (as is the network coding rate) is at most as difficult to preserve as the multicommodity flow rate.

Yet for the existential results on flow sparsifiers presented here, we automatically preserve the network coding rate (i.e. good *network coding sparsifiers* exist) - even though in general graphs we know nothing non-trivial about the relationship between flows and information. It remains an interesting open question to understand just how restrictive the assumption that a flow sparsifier be generated as a convex combination of 0-extensions is, as compared to the unrestricted question for which we prove lower bounds here.

8 Acknowledgments

We would like to thank Howard Karloff for many helpful discussions.

References

[1] M. Adler, N. Harvey, K. Jain, R. Kleinberg, and A. R. Lehman. On the capacity of information networks. *Symposium on Discrete Algorithms*, pages 251–260, 2006.

[2] A. Aggarwal and J. Vitter. The input/output complexity of sorting and related problems. *Communications of the ACM*, pages 116–127, 1988.

[3] R. Ahlswede, N. Cai, S. Y. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, pages 1204–1216, 2000.

[4] J. Batson, D. Spielman, and N. Srivastava. Twice-ramanujan sparsifiers. *Symposium on Theory of Computing*, pages 255–262, 2009.

[5] A. Benczúr and D. Karger. Approximating s-t minimum cuts in $\tilde{o}(n^2)$ time. *Symposium on Theory of Computing*, pages 47–55, 1996.

[6] A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. Woodruff. Transitive-closure spanners. *Symposium on Discrete Algorithms*, pages 932–941, 2009.

[7] D. Bienstock, E. Brickell, and C. Monma. On the structure of minimum-weight k-connected spanning networks. *SIAM Journal on Discrete Mathematics*, pages 320–329, 1990.

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

[9] Y. Chan, W. Fung, L. Lau, and C. Yung. Degree bounded network design with metric costs. *Foundations of Computer Science*, pages 125–134, 2008.

[10] P. Chew. There is a planar graph almost as good as the complete graph. *Symposium on Computational Geometry*, pages 169–177, 1986.

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

[12] R. Floyd. Permuting information in idealized two-level storage. *Complexity of Computer Calculations*, pages 105–109, 1972.

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

[14] A. Gupta, V. Nagarajan, and R. Ravi. Improved approximation algorithm for requirement cut. *submitted*, 2009.

[15] N. Harvey, R. Kleinberg, and A. R. Lehman. On the capacity of information networks. *IEEE Transactions on Information Theory*, pages 2345–2364, 2006.

[16] K. Jain, V. Vazirani, R. Yeung, and G. Yuval. On the capacity of multiple unicast sessions in undirected graphs. *IEEE International Symposium on Information Theory*, 2005.

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

[18] R. Khandekar, S. Rao, and U. Vazirani. Graph partitioning using single commodity flows. *Journal of the ACM*, 2009.

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

[20] G. Kramer and S. Savari. Edge-cut bounds on network coding rates. *Journal of Network and Systems Management*, 2006.

[21] Z. Li and B. Li. Network coding: The case of multiple unicast sessions. *Allerton Annual Conference on Communication, Control and Computing*, 2004.

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

[23] D. Shmoys. *Approximation algorithms for Cut problems and their application to divide-and-conquer* In *Approximation Algorithms for NP-hard Problems*. PWS, 1997.

[24] D. Spielman and N. Srivastava. Graph sparsification by effective resistances. *Symposium on Theory of Computing*, pages 563–568, 2008.

[25] D. Spielman and S. Teng. Nearly-linear time algorithms for graph partitioning, sparsification and solving linear systems. *Symposium on Theory of Computing*, pages 81–90, 2004.