SIAM J. COMPUT.
Vol. 43, No. 4, pp. 1239–1262

VERTEX SPARSIFIERS: NEW RESULTS FROM OLD TECHNIQUES*

MATTHIAS ENGLERT†, ANUPAM GUPTA‡, ROBERT KRAUTHGAMER§, HARALD RÄCKE¶, INBAL TALGAM-COHEN‖, AND KUNAL TALWAR#

Abstract. Given a capacitated graph $G = (V, E)$ and a set of terminals $K \subseteq V$, how should we produce a graph H only on the terminals K so that every (multicommodity) flow between the terminals in G could be supported in H with low congestion, and vice versa? (Such a graph H is called a *flow sparsifier* for G.) What if we want H to be a "simple" graph? What if we allow H to be a convex combination of simple graphs? Improving on results of Moitra [*Proceedings of the 50th IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Los Alamitos, CA, 2009, pp. 3–12] and Leighton and Moitra [*Proceedings of the 42nd ACM Symposium on Theory of Computing*, ACM, New York, 2010, pp. 47–56], we give efficient algorithms for constructing (a) a flow sparsifier H that maintains congestion up to a factor of $O(\frac{\log k}{\log\log k})$, where $k = |K|$; (b) a convex combination of trees over the terminals K that maintains congestion up to a factor of $O(\log k)$; (c) for a planar graph G, a convex combination of planar graphs that maintains congestion up to a constant factor. This requires us to give a new algorithm for the 0-extension problem, the first one in which the preimages of each terminal are connected in G. Moreover, this result extends to minor-closed families of graphs. Our bounds immediately imply improved approximation guarantees for several terminal-based cut and ordering problems.

Key words. approximation algorithms, vertex sparsifier, 0-extensions, planar graphs, graph minors, flow sparsifier, multicommodity flow, metric decomposition

AMS subject classifications. 68W25, 68W40, 68W20

DOI. 10.1137/130908440

1. Introduction.

Given an undirected capacitated graph $G = (V, E)$ and a set of terminal nodes $K \subseteq V$, we consider the question of producing a graph H only on the terminals K so that the congestion incurred on G and H for any multicommodity flow routed between terminal nodes is similar. Often, we will want the graph H to be structurally "simpler" than G as well. Such a graph H will be called a *flow sparsifier* for G; the *loss* (also known as *quality*) of the flow sparsifier is the factor by which the congestions in the graphs G and H differ. For instance, when $K = V$, the results of Räcke [31] give a convex combination of trees H with a loss of $O(\log n)$. We call this a *tree-based*

*Received by the editors February 4, 2013; accepted for publication (in revised form) April 14, 2014; published electronically July 3, 2014. A preliminary version appeared in the Proceedings of the 13th Workshop on Approximation Algorithms for Combinatorial Optimization Problems, Springer-Verlag, Berlin, 2010, pp. 152–165. A version of this work can also be found under http://arxiv.org/abs/1006.4586, 2010.

http://www.siam.org/journals/sicomp/43-4/90844.html

†Department of Computer Science and DIMAP, University of Warwick, Coventry, UK (englert@dcs.warwick.ac.uk). Supported by EPSRC grant EP/F043333/1 and DIMAP (the Centre for Discrete Mathematics and its Applications).

‡Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213 (anupamg@cs.cmu.edu). Research was partly supported by the NSF award CCF-0729022, and an Alfred P. Sloan Fellowship. This research was done when visiting Microsoft Research SVC, La Avenida, Mountain View CA.

§Weizmann Institute of Science, Rehovot, Israel (robert.krauthgamer@weizmann.ac.il). This work was supported in part by The Israel Science Foundation (grant 452/08), and by a Minerva grant.

¶Institut für Informatik, Technische Universität München, Munich, Germany (raecke@in.tum.de).

‖Computer Science Department, Stanford University, Stanford, CA 94350 (inbaltalgam@yahoo.com).

#Microsoft Research Silicon Valley, 1065 La Avenida, Mountain View, CA (kunal@microsoft.com).

flow sparsifier, meaning that it is a convex combination of trees.[1] Here and throughout, $k = |K|$ denotes the number of terminals, and $n = |V|$ the size of the graph.

For the case where $K \neq V$, it was shown by Moitra [29] and by Leighton and Moitra [26] that for every G and K, there exists a flow sparsifier $H = (K, E_H)$ whose loss is $O(\frac{\log k}{\log\log k})$, and moreover, one can efficiently (which means in polynomial time) find an $H' = (K, E_{H'})$ whose loss is $O(\frac{\log^2 k}{\log\log k})$. They used these to give approximation algorithms for several terminal-based problems, where the approximation factor depended polylogarithmically on the number of terminals k, and not on n. We note that they construct an arbitrary graph on K, and do not attempt to directly obtain "simple" graphs; e.g., to get tree-based flow sparsifiers on K, they apply to H' Räcke's method [31], and increase the loss by an $O(\log k)$ factor.

In this paper, we simplify and unify some of these results: we show that using the general framework of interchanging distance-preserving mappings and capacity-preserving mappings from [31], which was reinterpreted in an abstract setting by Andersen and Feige [1], we obtain the following improvements over the results of [29, 26].

1. We show that using the 0-extension results of [5, 12] in the framework of [31, 1] almost immediately gives us *efficent* constructions of flow sparsifiers with loss $O(\frac{\log k}{\log\log k})$. While the existential result of [26] also used the connection between 0-extensions and flow sparsifiers, the algorithmically efficient version of the result was done ab initio, increasing the loss by another $O(\log k)$ factor. We use existing machinery, thereby simplifying the exposition somewhat, and avoiding the increased loss. See Theorem 10.
2. We next use a randomized tree embedding due to [17], which is a variant of the so-called FRT tree embedding [13], where the expected stretch is reduced to $O(\log k)$ by requiring the noncontraction condition only for terminal pairs. Using this refined embedding in the framework of [31, 1], we obtain in Theorem 9 efficient constructions of *tree-based flow sparsifiers* with loss $O(\log k)$.
3. We then turn to special families of graphs. For planar graphs, we give a new 0-extension algorithm that outputs a convex combination of 0-extensions $f : V \to K$ (with $f(x) = x$ for all $x \in K$), such that all the corresponding 0-extension graphs $H_f = (K, E_f)$ (namely, $E_f = \{(f(u), f(v)) : (u, v) \in E\}$) are *planar graphs*, and its expected stretch $\max_{u,v\in V} \mathbb{E}[d_{H_f}(f(u),f(v))]/d_G(u,v) = O(1)$. In particular, the planar graphs H_f produced are graph-theoretic minors of G. These results are shown in section 4. We remark that the known 0-extension algorithms [5, 3, 24] do not ensure planarity of H_f.

   It follows that planar graphs admit a *planar-based flow sparsifier* (i.e., it is a convex combination of capacitated planar graphs on vertex set K) with loss $O(1)$, and that we can find these efficiently. The fact that flow sparsifiers with this loss *exist* was shown by [26], but their sparsifiers are not planar based.

   Moreover, the 0-extension algorithm itself can be viewed as a randomized version of Steiner point removal in metrics; previously, it was only known how to remove Steiner points from tree metrics with $O(1)$ distortion [16, 6]. We believe this randomized procedure is of independent interest; e.g., combined with an embedding of [18], this gives an alternate proof of the fact that

[1] More generally, for a class $\mathcal{F}$ of graphs, we define an $\mathcal{F}$-flow sparsifier to be a sparsifier that uses a single graph from $\mathcal{F}$, and an $\mathcal{F}$-based flow sparsifier to be a sparsifier that uses a convex combination of graphs from $\mathcal{F}$.

Table 1
Summary of our results. Previous results marked with † from [31], *all others from* [29, 26].

	Previous best result	Our result	Best result when $k = n$
Flow sparsifiers (efficient)	$O(\frac{\log^2 k}{\log\log k})$	$O(\frac{\log k}{\log\log k})$	—
Tree based flow sparsifiers	$O(\log n)^{\dagger}$, $O(\frac{\log^3 k}{\log\log k})$	$O(\log k)$	$\Theta(\log n)$
Minor-based flow sparsifiers	—	$O(\beta_G \log \beta_G)$	—
Steiner oblivious routing	$\widetilde{O}(\log^2 k)$	$O(\log k)$	$\Theta(\log n)$
ℓ-Multicut	$\widetilde{O}(\log^3 k)$	$O(\log k)$	$O(\log n)$
Steiner minimum linear arrangement (SMLA)	$\widetilde{O}(\log^{2.5} k)$	$O(\log k \log\log k)$	$O(\sqrt{\log n}\log\log n)$
SMLA in planar graphs	$\widetilde{O}(\log^{1.5} k)$	$O(\log\log k)$	$O(\log\log n)$
Steiner min cut linear arrangement	$\widetilde{O}(\log^4 k)$	$O(\log^2 k)$	$O(\log^{1.5} n)$
Steiner graph bisection	$O(\log n)^{\dagger}$, $O(\frac{\log^3 k}{\log\log k})$	$O(\log k)$	$O(\log n)$

the metric induced on the vertices of a single face of a planar graph can be embedded into a distribution over trees [25].

4. The results for planar graphs are in fact much more general. Suppose G is a β_G-decomposable graph (see definition in section 1.3). Then we can efficiently output a distribution over graphs $H_f = (K, E_f)$ such that these are all minors of G, and the expected stretch is
$$\max_{u,v\in V} \frac{\mathbb{E}[d_{H_f}(f(u), f(v))]}{d_G(u,v)} = O(\beta_G \log \beta_G).$$
Now applying the same ideas of interchanging distance and capacity preservation, given any G and K, we construct in Corollary 13 *minor-based flow sparsifiers* with loss $O(\beta_G \log \beta_G)$.
5. Finally, section 5 shows some lower bounds on flow sparsifiers; we show that flow sparsifiers that are 0-extensions of the original graph must have loss at least $\Omega(\sqrt{\log k})$ in the worst case. For this class of possible flow sparsifiers, this improves on the $\Omega(\log\log k)$ lower bound for sparsifiers proved in [26]. We also show that any flow sparsifier that only uses edge capacities which are bounded from below by a constant, must suffer a loss of $\Omega(\sqrt{\log k}/\log\log k)$ in the worst case.

We can use these results to improve the approximation ratios of several application problems (see section 6). In many cases, constructions based on trees allow us to use better algorithms. Our results are summarized in Table 1. Note that apart from the two linear-arrangement problems, our results smoothly approach the best known results for the case $k = n$.

Many of these applications further improve when the graph comes from a minor-closed family (and hence has good β-decompositions), e.g., for the Steiner minimum linear arrangement (SMLA) problem on planar graphs, we can get an $O(\log\log k)$-approximation by using our minor-based flow sparsifiers to reduce the problem to planar instances on the k terminals. Finally, in section 7 we show how to get better approximations for the Steiner linear arrangement problems above using direct linear programming (LP)/semidefinite programming (SDP) approaches.

1.1. Concurrent work. Concurrently and independently from our work, Charikar et al. [7] and independently Makarychev and Makarychev [28] gave an efficient construction for $O(\log k/\log\log k)$-quality flow sparsifiers. This is the same as our first result. Furthermore, Charikar et al. [7] give $O(\log k)$-quality tree-based flow sparsifiers, which is the same as our second result.

Makarychev and Makarychev [28] also consider the case of graphs that exclude a fixed minor. They make the existential result of Leighton and Moitra [26] constructive and provide $O(1)$-quality flow sparsifiers for these graphs. This is related to our third result. However, our construction has the additional advantage that the resulting flow sparsifiers are guaranteed to be graph-theoretic minors of the original graph. This, for instance, results in improved approximation guarantees for SMLA for planar graphs.

For cut sparsifiers, a weaker notion than flow sparsifiers [26], lower bounds of $\Omega(\sqrt[4]{\log k/\log\log k})$ and $\Omega(\sqrt[4]{\log k})$, were given by [28] and [7], respectively. (The former bound was improved to $\Omega(\sqrt{\log k}/\log\log k)$ in a later version.) Makarychev and Makarychev [28] show an additional lower bound of $\Omega(\sqrt{\log k/\log\log k})$ for flow sparsifiers, and also establish an interesting connection between flow and cut sparsifiers and Lipschitz extendability of maps in Banach spaces. Charikar et al. [7] also exhibit a family of graphs for which the (best possible) quality of cut sparsifiers with the restriction to 0-extensions is asymptotically larger than without such restriction.

1.2. Subsequent work. Subsequent to our work and using different techniques, Chuzhoy [9] shows that if the sparsifier H is allowed to contain a (relatively small) number of nonterminal vertices, it is possible to construct O(1)-quality cut sparsifiers of size $O(C^3)$ in time $n^{O(1)}\cdot 2^C$, and $O(1)$-quality flow sparsifiers of size $C^{O(\log\log C)}$ in time $n^{O(\log C)}\cdot 2^C$, where C is an upper bound on the sum of capacities of all edges incident to any single terminal. Andoni, Gupta, and Krauthgamer [2] obtained a flow sparsifier, of quality $1+\varepsilon$, which in effect is a trade-off between quality and size, for a restricted family that includes bipartite graphs.

Our results and techniques have proved useful in obtaining or simplifying other results. Lee, Mendel, and Moharrami [23] use our results to show an approximate version of the Okamura–Seymour theorem for node-capacitated graphs. Chekuri, Shepherd, and Weibel [8] study a problem similar to the Okamura–Seymour theorem, but with fewer restrictions on the demands. More specifically, they consider an undirected planar graph G and a set of demand pairs such that at least one vertex of each of the pairs lies in one of the outer k layers of G. They show that if, for any cut in G, the size of the cut is at least as large as the number of demand pairs that have exactly one vertex on each side of the cut, then the demands are integrally routable in G with congestion c^k for some universal constant c. Their proof also uses our results (unpublished note referenced in [8]).

Chuzhoy et al. [10] study the edge-connectivity k-route cut problem. In this problem an undirected edge-weighted graph, a set of demands consisting of pairs of vertices, and a number k are given. The goal is to compute a minimum-weight subset of edges such that removing these edges lets the edge connectivity of every demand pair drop below k. They give a polynomial-time bicriteria approximation of this problem which uses our algorithm for the ℓ-multicut problem as a building block to handle large values of k.

Recently, Kamma, Krauthgamer, and Nguyen [20] showed how to remove Steiner points from arbitrary graphical metrics (following the results of [16, 6] for tree metrics), and obtain a single minor of the input graph that achieves a polylogarithmic stretch (distortion) for all terminal-terminal distances. This result is not comparable to our result of $O(\beta_G \log \beta_G)$ expected stretch—our bound on the stretch is better, but our guarantee is only for the expected stretch for any fixed pair of terminals.

1.3. Notation. Our graphs will have edge lengths or capacities; all edge lengths will be denoted by $\ell : E \to \mathbb{R}_{\geq 0}$, and edge costs/capacities will be denoted by $c : E \to \mathbb{R}_{\geq 0}$. When we refer to a graph (G, ℓ), we mean a graph G with edge lengths $\ell(\cdot)$; similarly (H, c) denotes one with capacities $c(\cdot)$. When there is potential for confusion, we will add subscripts (e.g., $c_H(\cdot)$ or $\ell_G(\cdot)$) for disambiguation. Given a graph (G, ℓ), the shortest-path distances under the edge lengths ℓ is denoted by $d_G : V \times V \to \mathbb{R}_{\geq 0}$.

Given a graph $G = (V, E)$ and a subset of vertices $K \subseteq V$ designated as *terminals*, a *retraction* is a map $f : V \to K$ such that $f(x) = x$ for all $x \in K$. For (G, c) and terminals $K \subseteq V$, a *K-flow in G* is a multicommodity flow whose sources and sinks lie in K.

Decomposition of Metrics. Let (X, d) be a metric space. A partition (i.e., a set of disjoint "clusters") P of X is called *Δ-bounded* if every cluster $S \in P$ satisfies $\max_{u,v \in S} d(u, v) \leq \Delta$. The metric (X, d) is called *β-decomposable* if for every $\Delta > 0$ there is a polynomial-time algorithm to sample from a probability distribution μ over partitions of X, with the following properties.

- *Diameter bound:* Every partition $P \in \mathrm{supp}(\mu)$ is Δ-bounded.
- *Separation event:* For all $u, v \in X$, $\Pr_{P \in \mu}[\exists S \in P$ such that $u \in S$ but $v \notin S]$ $\leq \beta \cdot d(u, v)/\Delta$.

β-decompositions of metrics have become standard tools with many applications; for more information see, e.g., [24].

When the metric arises as the shortest-path distances d_G in a graph G with nonnegative edge lengths ℓ, we may assume that each cluster S in every partition P in the support of μ induces a *connected* subgraph of G; if not, break such a cluster into its connected components. The diameter bound and separation probabilities for edges remain unchanged by this operation; indeed, the diameter bound is obvious, and the separation probability for a nonadjacent pair (u, v) (and similarly when $d_G(u, v) < \ell_G(u, v)$) can be bounded by $\beta \cdot d_G(u, v)/\Delta$ by fixing a u-v shortest path and noting that for (u, v) to be separated, some shortest-path edge must be separated, and then applying the union bound.

We say that *a graph* $G = (V, E)$ is β-decomposable if for every assignment of nonnegative lengths ℓ to the edges, the resulting shortest-path metric d_G is β-decomposable.

2. 0-extensions. In this section we provide a definition of 0-extension which is somewhat different than the standard definition, and review some known results for 0-extensions. In Corollary 4, we also derive a variation of a known result on tree embeddings, which will be applied in section 3.2.

A 0-*extension* of graph $(G = (V, E), \ell_G)$ with terminals $K \subseteq V$ is usually defined as a retraction $f : V \to K$. We define a 0-extension to be a retraction $f : V \to K$ along with another graph $(H = (K, E_H), \ell_H)$; here, the length function $\ell_H : E_H \to \mathbb{R}_+$ is defined as $\ell_H(x, y) = d_G(x, y)$ for every edge $(x, y) \in E_H$. Note that this immediately implies $d_H(x, y) \geq d_G(x, y)$ for all $x, y \in K$. Note also that H_f defined in section 1 is a special case of H in which $E_H = \{(f(u), f(v)) : (u, v) \in E\}$, whereas, in general, H is allowed more flexibility (e.g., H can be a tree). This flexibility is precisely the reason we are interested both in the retraction f and in the graph H—we will often want H to be structurally simpler than G (just like we want a flow sparsifier to be simpler than the original graph).

For a (randomized) algorithm $\mathcal{A}$ that takes as input (G, ℓ_G) and outputs a (random) 0-extension (H, ℓ_H), the *stretch factor* of algorithm $\mathcal{A}$ is the minimum $\alpha \geq 1$

such that

$$\mathbb{E}_H[\, d_H(f(x), f(y))\,] \le \alpha\, d_G(x,y) \qquad \text{for all } x, y \in V.$$

The following are well-known results for 0-extension.

THEOREM 1 (see [12]). *There is an algorithm $\mathcal{A}_{FHRT}$ for* 0*-extension with stretch factor* $\alpha_{FHRT} = O(\frac{\log k}{\log\log k})$.

THEOREM 2 ([5]; see also [24]). *For graphs G that are β-decomposable, there is an algorithm $\mathcal{A}_{CKR}$ for* 0*-extension with stretch factor* $\alpha_{CKR} = O(\beta)$.

In particular, if the graph G belongs to a nontrivial family of graphs that is minor closed, it follows from [22, 14] that $\alpha = O(1)$.

2.1. 0-extension with trees. The following result is an extension of the tree-embedding theorem of Fakcharoenphol, Rao, and Talwar [13], where the difference is that the following result ensures the noncontracting property (property (a) of Theorem 3) only for terminal-terminal pairs, but replaces the $O(\log n)$ by $O(\log k)$ in the expected stretch between *any* pair of nodes. In what follows, a c-HST (abbreviation for hierarchically separated tree) is a rooted tree with edge lengths, that satisfies the following for some $D > 0$: the distance between every leaf and its ancestor at level $j \ge 0$ is exactly D/c^j. (As usual, level means hop distance from the root.)

THEOREM 3 (tree embedding [17]). *There is a randomized polynomial-time algorithm that takes as input a graph $G = (V, E)$ with terminals $K \subseteq V$ and outputs a (random) edge-weighted* 2*-HST $T = (I \cup L, E_T)$ with internal nodes I and leaves L, and a map $f : V \to L$, such that*

(a) *$d_T(f(x), f(y)) \ge d_G(x, y)$ for all $x, y \in K$ (with probability* 1*),*
(b) *$\mathbb{E}_T[d_T(f(x), f(y))] \le O(\log k)\, d_G(x, y)$ for all $x, y \in V$, and*
(c) *for each nonterminal $v \in V \setminus K$, either there exists a terminal x_v sharing the leaf node with it (i.e., $f(v) = f(x_v)$), or another descendent of $f(v)$'s parent in T contains a terminal x_v.*

COROLLARY 4 (tree 0-extension). *There is a randomized polynomial-time algorithm $\mathcal{A}_{GNR}$ for* 0*-extension that has stretch factor $\alpha_{GNR} = O(\log k)$; furthermore, the graphs output by the algorithm are trees on the vertex set K.*

Proof. We need to give an algorithm that takes as input a graph $G = (V, E)$ with terminals $K \subseteq V$ and outputs a (random) edge-weighted tree $T = (K, E)$ and a retraction $f : V \to K$ such that

(a′) $d_T(x, y) \ge d_G(x, y)$ for all $x, y \in K$ (with probability 1),
(b′) $\mathbb{E}_T[d_T(f(x), f(y))] \le O(\log k)\, d_G(x, y)$ for all $x, y \in V$.

We may assume that in G, all terminals are at nonzero distance from each other; otherwise, we can remove some terminals (from K, without changing G), apply the proof below, and add the terminals back in at the end.

We start with sampling from the distribution of Theorem 3 a random tree $T' = (I \cup L, E')$ and an associated map f. We can take any leaf $l \in L$ whose preimage set only contains nonterminals, remove the leaf, and remap all $v \in f^{-1}(l)$ to some other leaf that is a descendent of l's parent node and also contains a terminal. (Such a leaf is guaranteed to exist by property (c) of Theorem 3.) While both the tree and the map change, we continue to call the modified tree T' and the map f. We repeat this process until all leaves in the modified tree T' contain at least one terminal. Now property (a) implies (recall that in G, the distances between all terminals were nonzero) that each leaf contains at most one terminal. Hence $f|_K$ is a 1-1 correspondence between the terminal set K and the remaining leaves in the tree T'. Since the tree T' is a 2-HST, the distances in the tree between a remapped nonterminal and any other node in T' (apart from the one it was identified with) do not change.

We can now remove all internal nodes in the modified version of T' (using, say, [16]) to get a tree $T'' = (L, E'')$ on just the (erstwhile) leaves such that none of the $f(u)$-$f(v)$ distances are shrunk, and they are stretched by a factor of at most 8. The bijection between the set L and terminals K allows us to view the tree T'' as being on the node set K, and the map f as being a retraction from $V \to K$. Finally, shrinking the edges of the tree T'' only makes the expected stretch smaller, so we can reduce the length of any tree edge $e = (x, y)$ in T'' and set it equal to $d_G(x, y)$. Call this final tree T; it is immediate from properties (a) and (b) that this random T and the associated retraction $f : V \to K$ satisfy properties (a$'$) and (b$'$) above, where the O term in property (b$'$) hides an extra stretch of 8 due to this postprocessing. ∎

As an aside, a weaker version of Corollary 4 with $O(\frac{\log^2 k}{\log \log k})$ can be proved as follows. First use Theorem 1 to obtain a random 0-extension H from G such that $\mathbb{E}_H[d_H(x, y)] \le O(\frac{\log k}{\log \log k})\ d_G(x, y)$ for all $x, y \in K$. Then use the result of [13] to get a random tree $H' = (K, E_{H'})$ such that $\mathbb{E}_{H'}[d_{H'}(x, y)] \le O(\log k)\ d_H(x, y)$ for all $x, y \in V(H)$. Combining these two results proves the weaker claim.

3. Flow sparsifiers via 0-extensions. In this section we first present the general framework of interchanging distance-preserving mappings and capacity-preserving mappings from [31], and its more abstract interpretation by Andersen and Feige [1], and then discuss an algorithmically efficient implementation of it. We then apply this framework, and "transfer" the results of section 2, which are aimed at preserving distances, to results about preserving capacities, which are essentially constructions of flow sparsifiers.

Recall that given an edge-capacitated graph (G, c) and a set $K \subseteq V$ of terminals, a *flow sparsifier with quality* $\rho \ge 1$ is another capacitated graph $(H = (K, E_H), c_H)$ such that (a) any feasible K-flow in G can be feasibly routed in H, and (b) any feasible K-flow in H can be routed in G with congestion ρ.

3.1. Interchanging distance and capacity. 3We now use the framework of Räcke [31], as interpreted by Andersen and Feige [1]. Given a graph $G = (V, E)$, let $\mathcal{P}$ be a collection of multisets of E, which will henceforth be called *paths*. A mapping $M : E \to \mathcal{P}$ maps each edge e to a path $M(e)$ in $\mathcal{P}$. Such a map can be represented as a matrix $\mathbf{M}$ in $\mathbb{Z}^{E \times E}$, where $\mathbf{M}_{e,e'}$ is the number of times the edge e' appears in the path (multiset) $M(e)$. Given a collection $\mathcal{M}$ of mappings (which we call the *admissible* mappings), a *probabilistic mapping* is a probability distribution over (or, convex combination of) admissible mappings; i.e., define $\lambda_M \ge 0$ for each $M \in \mathcal{M}$ such that $\sum_{M \in \mathcal{M}} \lambda_M = 1$.

Distance mappings. Given a graph $G = (V, E)$ with edge lengths $\ell : E \to \mathbb{R}_{>0}$,

- the *stretch* of an edge $e \in E$ under a mapping M is $\sum_{e'} \mathbf{M}_{e,e'} \ell(e')/\ell(e)$;
- the *average stretch* of e under a probabilistic mapping $\{\lambda_M\}$ is

$$\sum_M \lambda_M \left(\sum_{e'} \mathbf{M}_{e,e'} \ell(e')/\ell(e) \right);$$

- the *stretch of a probabilistic mapping* is the maximum over all edges of their average stretch.

Capacity mappings. Given a graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}_{>0}$,

- the *load* of an edge $e' \in E$ under a mapping M is $\sum_e \mathbf{M}_{e,e'} c(e)/c(e')$;

- the *expected load* of e' under a probabilistic mapping $\{\lambda_M\}$ is

$$\sum_M \lambda_M \left(\sum_e \mathbf{M}_{e,e'} c(e)/c(e') \right);$$

- the *congestion of a probabilistic mapping* is the maximum over all edges of their expected loads.

The transfer theorem. Andersen and Feige [1] distilled ideas from Räcke [31] to state the following theorem.

THEOREM 5 (see [1, Theorem 6]). *Fix a graph $G = (V, E)$ and a collection $\mathcal{M}$ of admissible mappings. For every $\rho \geq 1$, the following are equivalent:*

1. *For every collection of edge lengths $\ell(\cdot)$, there is a probabilistic mapping with stretch at most ρ;*
2. *for every collection of edge capacities $c(\cdot)$, there is a probabilistic mapping with congestion at most ρ.*

Andersen and Feige [1] also outline how to make this result algorithmic: if one can efficiently sample from the probabilistic distance mapping with stretch ρ (which is true for the settings in this paper), one can efficiently sample from a probabilistic capacity mapping with congestion $O(\rho)$ (and *vice versa*). In fact, one can obtain an explicit distribution on polynomially many admissible mappings. The techniques of Räcke [31] can also be used to obtain this algorithmic version of the transfer theorem. Merely for completeness, in the following we show how to derive the algorithmic result from a special case of a theorem by Khandekar [21].

THEOREM 6 (see [21, Theorem 5.1.6]). *Let $P \subseteq \mathbb{R}^d$ be a nonempty convex set for some d, and for each $e \in E$, let $f_e : P \to \mathbb{R}_{\geq 0}$ be a nonnegative continuous convex function. Suppose we have an oracle that, given a vector $x \in \mathbb{R}^E_{\geq 0}$ with $\sum_{e\in E} x_e = 1$, finds $\lambda \in P$ such that $\sum_{e\in E} x_e f_e(\lambda) \leq \rho$. Then there exists an algorithm that given an error parameter $\omega \in (0, 1)$ computes $\lambda \in P$ such that $\max_{e\in E} f_e(\lambda) \leq e^{\omega}\rho$, while making $O(\omega^{-2} m \log m)$ calls to the oracle and an equal number of evaluations of $f_e(\cdot)$, where $m = |E|$.*

This theorem can be used to show the following algorithmic version of the transfer theorem.

COROLLARY 7. *Fix a graph $G = (V, E)$ and a collection $\mathcal{M}$ of admissible mappings. For every $\rho \geq 1$ and constant $\omega \in (0, 1)$ we have the following.*

(a) *Suppose that for every collection of edge lengths $\ell(\cdot)$ (edge capacities $c(\cdot)$) there is an efficient algorithm to* compute *a probabilistic mapping with stretch (congestion) at most ρ. Then for every collection of edge capacities $c(\cdot)$ (edge lengths $\ell(\cdot)$) there exists an efficient algorithm to compute a probabilistic mapping with congestion (stretch) at most $e^{\omega}\rho$.*

(b) *Suppose that for every collection of edge lengths $\ell(\cdot)$ (edge capacities $c(\cdot)$) there is an efficient algorithm to* sample *from a probabilistic mapping with stretch (congestion) at most ρ. Then for every collection of edge capacities $c(\cdot)$ (edge lengths $\ell(\cdot)$) there exists an efficient algorithm to compute a probabilistic mapping whose congestion (stretch) is, with high probability and in expectation, at most $e^{2\omega}\rho + 1$.*

Proof. We will show how to obtain a low-congestion probabilistic mapping if we can, for every collection of edge lengths $\ell(\cdot)$, efficiently compute (or sample from) a probabilistic mapping with low stretch. The other direction, i.e., obtaining low stretch when we have a method to obtain low-congestion probabilistic mappings, is symmetric.

(a) We define $f_{e'}(\lambda) := \sum_M \lambda_M (\sum_e \mathbf{M}_{e,e'} c(e)/c(e'))$ to be the expected load of an edge $e' \in E$ under probabilistic mapping $\{\lambda_M\}$ and we choose P to be the set of all nonnegative $|\mathcal{M}|$-dimensional vectors λ with $\sum_{M \in \mathcal{M}} \lambda_M = 1$. Now Theorem 6 immediately implies the claim if we can implement the oracle efficiently.
Define edge lengths $\ell(e) := x_e / c(e)$. Due to our assumption, we can efficiently find a probabilistic mapping $\{\lambda_M\}$ such that the maximum average stretch, with respect to these edge lengths, is at most ρ, i.e., such that

$$\max_e \sum_M \lambda_M \left(\sum_{e'} \mathbf{M}_{e,e'} \frac{\ell(e')}{\ell(e)} \right) \leq \rho \ .$$

Plugging in $\ell(\cdot)$, we obtain

$$\max_e \sum_M \lambda_M \left(\sum_{e'} \mathbf{M}_{e,e'} \frac{\ell(e')}{\ell(e)} \right) = \max_e \frac{1}{x_e} \sum_M \left(\lambda_M \sum_{e'} x_{e'} \cdot \mathbf{M}_{e,e'} \frac{c(e)}{c(e')} \right) \leq \rho \ .$$

Therefore, we can find $\{\lambda_M\}$ such that, for every e,

$$\sum_M \lambda_M \left(\sum_{e'} x_{e'} \cdot \mathbf{M}_{e,e'} c(e)/c(e') \right) \leq \rho \cdot x_e.$$

Summing up over all e gives $\sum_e \sum_M \lambda_M (\sum_{e'} x_{e'} \mathbf{M}_{e,e'} c(e)/c(e')) \leq \rho \sum_e x_e = \rho$ and hence, by rearranging the sums,

$$\sum_{e'} x_{e'} \left(\sum_M \lambda_M \left(\sum_e \mathbf{M}_{e,e'} \frac{c(e)}{c(e')} \right) \right) = \sum_{e'} x_{e'} f_{e'}(\lambda) \leq \rho \ .$$

This completes the implementation of the oracle.

(b) Above we assumed that we can efficiently *compute* an explicit distribution on polynomially many admissible mappings that results in a probabilistic mapping with low stretch. If we can only efficiently *sample* from such a distribution $\{\lambda_M\}$, we can still obtain a similar result. Let C be an upper bound on the worst load of any edge under any admissible mapping (e.g., the maximum sum of all entries of an $M \in \mathcal{M}$ multiplied by the largest ratio of capacities of two different edges). Then, for a sufficiently large constant κ we take $T = \ln(mC/\omega) \cdot \kappa/\omega$ independent samples from $\{\lambda_M\}$ and pick the sampled $M' \in \mathcal{M}$ that minimizes $\sum_{e'} x_{e'} (\sum_e \mathbf{M'}_{e,e'} c(e)/c(e'))$. Our oracle then returns λ' with $\lambda'_{M'} = 1$ (and $\lambda'_{M''} = 0$ for all $M'' \neq M'$).
For a single sample, the probability that $\sum_{e'} x_{e'} f_{e'}(\lambda') > e^{\omega}\rho$ is at most $1/e^{\omega}$ due to Markov's inequality. The probability that this is the case for all T independent samples is at most $1/e^{\omega T} = (mC/\omega)^{-\kappa}$. By taking a union bound over all $O(\omega^{-2} m \log m)$ oracle calls we conclude that the probability that any of them returns λ' with $\sum_{e'} x_{e'} f_{e'}(\lambda') > e^{\omega}\rho$ is bounded by $O((mC)^{2-\kappa})$. Therefore, Theorem 6 guarantees that with high probability, namely, with probability at least $1 - O((mC)^{2-\kappa})$, we obtain a $\gamma \in P$ with $\max_e f_e(\gamma) \leq e^{2\omega}\rho$. With the remaining probability $O((mC)^{2-\kappa})$, $\max_e f_e(\gamma)$ may be much larger, but even in the worst case it will be bounded by C. Therefore, by choosing κ sufficiently large, the expectation of $\max_e f_e(\gamma)$ is bounded by $e^{2\omega}\rho + C \cdot O((mC)^{2-\kappa}) \leq e^{2\omega}\rho + 1$. $\square$

3.2. Constructing sparsifiers. The following theorem gives the formal connection between 0-extensions and flow sparsifiers.

THEOREM 8. *Suppose there is a (randomized) algorithm A that, given a graph G and edge lengths $\ell_G : E(G) \to \mathbb{R}^+$, computes a 0-extension $((H, \ell_H), f)$ with stretch factor at most α such that H is a graph from class $\mathcal{H}$.*

Then there is an algorithm that, given any capacity assignment $c_G : E(G) \to \mathbb{R}^+$, computes for the graph (G, c_G) an $O(\alpha)$-loss flow sparsifier that is a convex combination of edge-capacitated graphs from class $\mathcal{H}$.

Proof. Suppose we have a 0-extension (H, f), where $H = (K, E_H)$ and $f : V \to K$ is a retraction. For every pair of terminals $u, v \in K$ we fix a canonical shortest path $S^H_{u,v}$ between u and v in H and a canonical shortest path $S^G_{u,v}$ between u and v in G (observe that for the important case that $\mathcal{H}$ is the set of trees the paths in H are unique). We define a mapping $M_{H,f} : E(G) \to \mathcal{P}$ corresponding to 0-extension (H, f) by

$$M_{H,f}((x,y)) = \biguplus_{(u,v)\in S^H_{f(x)f(y)}} S^G_{uv} \ .$$

In other words an edge (x, y) is first mapped to $S^H_{f(x)f(y)}$ in H and then the edges (u, v) on this path are mapped to path S^G_{uv} in G. Recall that $M_{H,f}((x, y))$ is a multi-set. In the corresponding matrix representation, $\mathbf{M}_{e,e'}$ is the multiplicity of e' in the set $\biguplus_{(u,v)\in S^H_{f(x)f(y)}} S^G_{uv}$.

For a graph class $\mathcal{H}$ (for example the set of trees) we define the set of *admissible* mappings by $\{M_{H,f} \mid H \in \mathcal{H}\}$. Note that in $M_{H,f}$ an edge $(x, y) \in E(G)$ is mapped to a path of length $d_H(f(x), f(y))$. This means the stretch of the edge in the mapping is the same as the stretch of an edge in the definition of 0-extensions. Therefore, the existence of a probability distribution over 0-extensions with (expected) stretch α gives rise to a probability distribution over admissible mappings with (expected) stretch α.

Applying the constructive version of the transfer theorem gives that for any assignment $c_G : E(G) \to \mathbb{R}^+$ of edge capacities to edges in G, we can compute a probability distribution over admissible mappings with congestion at most $O(\alpha)$. In the following we show that we can interpret this probability distribution as a flow sparsifier.

With every mapping $M_{H,f}$ we associate the graph H with the following edge capacities

$$c_{H,f}(e) = \sum_{(u,v)\in E(G): e\in S^H_{f(u),f(v)}} c_G((u, v)) \ .$$

This means the capacity of an edge $e \in E(H)$ is the total capacity of all graph edges $(u, v) \in G$ for which the canonical path between u and v in H contains e. The flow sparsifier F is now the convex combination $\{\lambda_{H,f}\}$ over graphs $(H, c_{H,f})$. To see that F has quality $O(\alpha)$ we prove two facts:

(a) Any K-flow that can be feasibly routed in G, can also be feasibly routed in F; and

(b) any K-flow that can be feasibly routed in F, can be routed with congestion $O(\alpha)$ in G.

Proving these facts is essentially a matter of unraveling the definitions. For (a), the definition of edge capacities $c_{H,f}$ ensures that $(H, c_{H,f})$ can feasibly route all edges of

G concurrently. Hence, it can also route any K-flow that is feasible in G. Since this is true for any graph $(H, c_{H,f})$ it also holds for the convex combination F.

To prove (b), we want to route edges of F in G. As F is a convex combination this means we want to concurrently route all graphs $(H, c_{H,f})$, where the capacities are scaled down by the convex multiplier $\lambda_{H,f}$. We simply route an edge $(u, v) \in H$ along the canonical path S^G_{uv}. This results in the following load on an edge $e' \in E(G)$:

$$\frac{1}{c(e')} \sum_{H,f} \lambda_{H,f} \sum_{e_H=(u,v)\in E(H): e' \in S^G_{u,v}} c_{H,f}(e_H) \ .$$

Plugging in the definition for the edge capacities $c_{H,f}$ and changing the order of summation gives that this is equal to

$$\frac{1}{c(e')} \sum_{H,f} \lambda_{H,f} \sum_{e_H=(u,v)\in E(H): e' \in S^G_{uv}} \ \sum_{(x,y)\in E(G): e_H \in S^H_{f(x),f(y)}} c(xy)$$

$$= \frac{1}{c(e')} \sum_{H,f} \lambda_{H,f} \sum_{(x,y)\in E} c(xy) \cdot (\text{multiplicity of } e' \text{ in } \uplus_{(u,v)\in S^H_{f(x),f(y)}} S^G_{uv}) \ .$$

However, this is exactly the *expected load* for e' under the notion of admissible maps defined in (3.2); hence this is bounded by the congestion (the maximum expected load over all edges), which is at most $O(\alpha)$. This proves condition (b) above, that the congestion to route any K-flow in the convex combination F in the graph G is at most $O(\alpha)$. ∎

Combining Theorem 8 with Corollary 4 gives the following.

THEOREM 9 (tree-based flow sparsifiers). *There is a randomized polynomial-time algorithm that, given a graph G and terminals K, outputs a flow sparsifier H which is a convex combination of trees and has loss $O(\log k)$.*

Combining Theorem 8 with Theorem 1 gives the following.

THEOREM 10 (flow sparsifiers). *There is a randomized polynomial-time algorithm that, given a graph G with terminals K, outputs a flow sparsifier H with loss $O(\frac{\log k}{\log \log k})$.*

The same idea using 0-extension results for β-decomposable graphs (Theorem 2) gives us the following.

THEOREM 11 (flow sparsifiers for minor-closed families). *There is a randomized polynomial-time algorithm that, given a β-decomposable graph G with terminals K, constructs a flow sparsifier with loss $O(\beta)$.*

Note that the decomposability holds if G belongs to a nontrivial minor-closed-family $\mathcal{G}$ (e.g., if G is planar). However, Theorem 11 does not claim that the flow sparsifier for G also belongs to the family $\mathcal{G}$; this is the question we resolve in the next section.

4. Connected 0-extensions and minor-based flow sparsifiers. The results in this section apply to β-decomposable graphs. A prominent example of such graphs are planar graphs, which (along with every family of graphs excluding a fixed minor) are $O(1)$-decomposable [22, 14]. Thus, Theorem 12, Corollary 13, and Theorem 14 below all apply to planar graphs (and more generally to excluded-minor graphs) with $\beta = O(1)$. We now state our results for β-decomposable graphs in general. In section 4.2 we define a related notion called *terminal decomposability*, and show analogous results for $\hat{\beta}$-terminal-decomposable graphs.

In what follows we use the definition of 0-extension from section 2 with $H = H_f$, i.e., $E_H = \{(f(u), f(v)) : (u, v) \in E\}$, hence the 0-extension is completely defined by the retraction f. We say that a 0-extension f is *connected* if for every x, $f^{-1}(x)$ induces a connected component in G. Our main result shows that we get connected 0-extensions with stretch $O(\beta \log \beta)$ for β-decomposable metrics.

THEOREM 12 (connected 0-extension). *There is a randomized polynomial-time algorithm that, given $(G = (V, E), \ell_G)$ with terminals K such that d_G is β-decomposable, produces a connected 0-extension $f : V \to K$ such that for all $u, v \in V$, we have*

$$\mathbb{E}[d_H(f(u), f(v))] \leq O(\beta \log \beta) \cdot d_G(u, v).$$

Note that if f is a connected 0-extension, the graph H_f is a minor of G. Applying Theorem 5 to interchange the distance preservation with capacity preservation, we get the following analogue of Theorem 9.

COROLLARY 13 (minor-based flow sparsifiers). *For every β-decomposable graph $G = (V, E)$ with edge capacities c_G and a subset $K \subset V$ of k terminals, there is a minor-based flow sparsifier with quality $O(\beta \log \beta)$. Moreover, a minor-based flow sparsifier for G, c_G, K can be computed efficiently in randomized polynomial-time.*

Since planar graphs are $O(1)$-decomposable and since their minors are planar, by Corollary 13 they have an efficiently constructable planar-based flow sparsifier with quality $O(1)$. By Theorem 12, they always have a connected 0-extension with stretch at most $O(1)$. An interesting consequence of the latter result is that given any planar graph (G, ℓ_G), and a set K of terminals, we can "remove" the nonterminals and get a related planar graph on K while preserving interterminal distances in expectation. Moreover, this extends to every family of graphs excluding a fixed minor. These results generalize a result from [16] showing a similar result for trees.[2]

THEOREM 14 (Steiner points removal). *There is a randomized polynomial-time algorithm that, given $(G = (V, E), \ell_G)$ and K such that d_G is β-decomposable, outputs minors $H = (K, E_H)$ of G such that $1 \leq \frac{\mathbb{E}[d_H(x,y)]}{d_G(x,y)} \leq O(\beta \log \beta)$ for all $x, y \in K$.*

Note that, since general graphs are only $\Theta(\log n)$-decomposable, these results only give us an $O(\log n \log \log n)$-approximation for connected 0-extension on arbitrary graphs (or an $O(\log^2 k \log \log k)$-approximation using results of section 4.2). We can improve that to $O(\log k)$; the details are in section 4.3.

THEOREM 15 (connected Calinescu–Karloff–Rabani (CKR)). *There is a randomized polynomial-time algorithm that on input $(G = (V, E), \ell_G)$ and K, produces a connected 0-extension f with $\mathbb{E}[d_H(f(u), f(v))] \leq O(\log k) \cdot d_G(u, v)$ for all $u, v \in V$.*

Using the semimetric relaxation for 0-extension, we get a connected 0-extension whose cost is at most $O(\log k)$ times the optimal (possibly disconnected) 0-extension. To our knowledge, this is the first approximation algorithm for connected 0-extension, and in fact shows that the gap between the optimum connected 0-extension and the optimum 0-extension is bounded by $O(\log k)$. The same is true with an $O(1)$ bound for planar graphs. We remark that the connected 0-extension problem is a special case of the connected metric labeling problem, which has recently received attention in the vision community [33, 30].

4.1. The algorithm for decomposable metrics. We now give the algorithm behind Theorem 12. Assume that edge lengths ℓ_G are integral and scaled such that

[2]One difference from the result in [16] is the following: that result deterministically produced a single tree after removing the nonterminals, and hence the distances were preserved deterministically, and not just in expectation. Getting such a result for planar graphs remains an open problem.

the shortest edge is of length 1. Let the diameter of the metric be at most 2^δ. For each vertex $v \in V$, define $A_v = \min_{x \in K} d_G(v, x)$ to be the distance to the closest terminal. The algorithm maintains a partial mapping f at each point in time—some of the $f(v)$'s may be undefined (denoted by $f(v) = \perp$) during the run, but f is a well-defined 0-extension when the algorithm terminates. We say a vertex $v \in V$ is *mapped* if $f(v) \neq \perp$. The algorithm appears as Algorithm 1.

Algorithm 1. Algorithm for connected 0-extension.

input: $(G, \ell_G), K$.
let $i \leftarrow 0$, $f(x) = x$ for all $x \in K$, $f(v) = \perp$ for all $v \in V \setminus K$.
while there is a v such that $f(v) = \perp$ **do**
    let $i \leftarrow i + 1$, $r_i \leftarrow 2^i$
    sample a β-decomposition of d_G with diameter bound r_i to get a partition P
    for all clusters C_s in the partition P that contains both mapped and unmapped vertices **do**
        delete all vertices u in C_s with $f(u) \neq \perp$
        for each connected component C from C_s **do**
            choose a vertex $w_C \in C_s$ that was deleted and had an edge to C
            reset $f(u) = f(w_C)$ for all $u \in C$.
        end for
    end for
end while

We can assume that in round $\delta = \log \operatorname{diam}(G)$, the partitioning algorithm returns a single cluster, in which case all vertices are mapped and the algorithm terminates. Let f_i be the mapping at the end of iteration i. For $x \in K$, let V_i^x denote $f_i^{-1}(x)$, the set of nodes mapped to x. The following claim follows inductively:

Lemma 16. *For every iteration i and $x \in K$, the set V_i^x induces a connected component in G.*

Proof. We prove the claim inductively. For $i = 0$, there is nothing to prove since $V_i^x = \{x\}$. Suppose that in iteration i, we map vertex u to x so that $u \in V_i^x$. Thus for some component C containing u, the mapped neighbor w_C chosen by the algorithm was in V_{i-1}^x. Since we map all of C to x, there is a path connecting v to w_C in V_i^x. Inductively, w_C is connected to x in $V_{i-1}^x \subseteq V_i^x$, and the claim follows. ∎

The following lemma will be useful in the analysis of the stretch; it says that any node mapped in iteration i is mapped to a terminal at distance $O(2^i)$.

Lemma 17. *For every iteration i and $x \in K$, and every $u \in V_i^x$, $d_G(x, u) \leq 2r_i$.*

Proof. The proof is inductive. For $i = 0$, the claim is immediate. Suppose that in iteration i, we map vertex u to x so that $u \in V_i^x$. Thus for some component C containing u, the mapped neighbor w_C chosen by the algorithm was in V_{i-1}^x. Moreover, u and w_C were in the same cluster in the decomposition so that $d(u, w_C) \leq r_i$. Inductively, $d(w_C, x) \leq 2r_{i-1}$ and the claim follows by triangle inequality. ∎

In the remainder of the section, we bound the stretch of the 0-extension; for every edge $e = (u, v)$ of G, we show that

$$\mathbb{E}[d_G(f(u), f(v))] \leq O(\beta \log \beta)\, d_G(u, v).$$

Note that for $e = (u, v)$, $d_G((f(u), f(v)) = d_H((f(u), f(v))$. Therefore it is sufficient to prove the claim for d_G. The analogous claim for nonadjacent pairs will follow by triangle inequality, but here with d_H. We say that the edge $e = (u, v)$ is *settled in round j* if the latter of its endpoints gets mapped in this round; e is *untouched after* *round* j if both u and v are unmapped at the end of round j. Let $d_G(u,K) \le d_G(v,K)$ and let A_e denote the distance $d_G(u,K)$. Let $j_e := \lfloor \log(A_e) \rfloor - 1$.

LEMMA 18. *For edge* $e = (u,v)$,

(a) *edge* e *is untouched after round* $j_e - 1$,

(b) *if edge* e *is settled in round* j *then* $d_G(f(u), f(v)) = O(2^j + d_G(u,v))$.

Proof. For (a), if one of the endpoints of e is mapped before round j_e, then $2 \cdot 2^{j_e} \le A_e = d_G(e,K)$, which contradicts Lemma 17. For (b), both $d_G(u, f(u)), d_G(v, f(v)) \le 2^{j+1}$ by Lemma 17; the triangle inequality completes the proof. $\square$

Let $\mathcal{B}_j$ denote the "bad" event that the edge is settled in round j and that both endpoints are mapped to different terminals. Let $z := \max\{A_e, d_G(u,v)\}$. We want to use

$$\mathbb{E}[d(f(u), f(v))] = \sum_j \Pr[\mathcal{B}_j] \cdot \mathbb{E}[d(f(u), f(v)) \mid \mathcal{B}_j].$$

CLAIM 19. $\Pr[\mathcal{B}_j] \le \min\{4\beta \frac{z}{2^j}, 1\} \cdot 5\beta \frac{d_G(u,v)}{2^j}$.

Proof. Recall that an edge is untouched after round j' if neither of its endpoints is mapped at the end of this round. For this to happen, u must be separated from its closest terminal in the clustering in round j', which happens with probability at most $\min\{\beta \frac{A_e}{2^{j'}}, 1\}$. Also recall that the probability that an edge $e = (u,v)$ is cut in a round j' is at most $\beta \frac{d_G(u,v)}{2^{j'}}$. Let i denote the round in which the edge is first touched. We upper bound the probability of the event $\mathcal{B}_j$ separately depending on how i and j compare. Note that for $j \le 2$, the right-hand side is at least 1 so the claim holds trivially.

- $i \le j-2$. For $\mathcal{B}_j$ to occur, the edge e must be cut in round $j-2$ and $j-1$, as otherwise it would already be settled in one of these rounds. The probability of this is at most $\min\{\beta \frac{d_G(u,v)}{2^{j-2}}, 1\} \cdot \beta \frac{d_G(u,v)}{2^{j-1}} \le \min\{4\beta \frac{z}{2^j}, 1\} \cdot 2\beta \frac{d_G(u,v)}{2^j}$.
- $i = j-1$. For $\mathcal{B}_j$ to occur, the edge e must be cut in round $j-1$ and must be untouched after round $j-2$. The probability of this is at most $\min\{\beta \frac{A_e}{2^{j-2}}, 1\} \cdot \beta \frac{d_G(u,v)}{2^{j-1}} \le \min\{4\beta \frac{z}{2^j}, 1\} \cdot 2\beta \frac{d_G(u,v)}{2^j}$.
- $i = j$. For $\mathcal{B}_j$ to occur, e must be cut in round j and must be untouched after round $j-1$. The probability of this is at most $\min\{\beta \frac{A_e}{2^{j-1}}, 1\} \cdot \beta \frac{d_G(u,v)}{2^j} \le \min\{4\beta \frac{z}{2^j}, 1\} \cdot \beta \frac{d_G(u,v)}{2^j}$.

Since $\Pr[\mathcal{B}_j] = \Pr[\mathcal{B}_j \wedge (i \le j-2)] + \Pr[\mathcal{B}_j \wedge (i = j-1)] + \Pr[\mathcal{B}_j \wedge (i = j)]$, the claim follows. $\square$

Lemma 18(b) implies that if the edge is settled before round $j_d := \lfloor \log(d_G(u,v)) \rfloor$, the conditional expectation $\mathbb{E}[d_G(f(u), f(v)) \mid \mathcal{B}_j]$ is $O(d_G(u,v))$. Moreover the edge e cannot be settled before round $j_e = \lfloor \log(A_e) \rfloor - 1$ by Lemma 18(a). Let $j_m := \max\{j_d, j_e\}$. It therefore suffices to show that

$$\sum_{j \ge j_m} \Pr[\mathcal{B}_j] \cdot O(2^j) \le O(\beta \log \beta)\, d_G(u,v).$$

Plugging in the upper bound for $\Pr[\mathcal{B}_j]$ into the left-hand side, we get

$$\begin{aligned}
\sum_{j \ge j_m} \Pr[\mathcal{B}_j] \cdot O(2^j) &\le \sum_{j \ge j_m} \min\{4\beta \tfrac{z}{2^j}, 1\} \cdot 5\beta \tfrac{d_G(u,v)}{2^j} \cdot O(2^j) \\
&\le \sum_{j \ge j_m} \min\{4\beta \tfrac{z}{2^j}, 1\} \cdot \beta \cdot O(d_G(u,v)) \\
&\le O(\beta \log \beta)\, d_G(u,v).
\end{aligned}$$

In the last step, we used that $z = \max\{A_e, d_G(u,v)\} \le \max\{2^{j_e+2}, 2^{j_d+1}\} \le 2^{j_m+2}$, so the first $O(\log\beta)$ terms contribute $O(\beta\, d_G(u,v))$, while the remaining terms form a geometric series and sum to $O(d_G(u,v))$. This completes the proof of Theorem 12.

4.2. Terminal decompositions. The general theorem for connected 0-extensions gives a guarantee in terms of its decomposition parameter β, and in general this quantity may depend on n. This seems wasteful, since we decompose the entire metric while we mostly care about separating the terminals.

To this end, we define *terminal decompositions* (the reader might find it useful to contrast it with the definition of decompositions in section 1.3). A *partial partition* of a set X is a collection of disjoint subsets (called "clusters" of X). A metric (X,d) with terminals K is called *$\hat\beta$-terminal decomposable* if for every $\Delta > 0$ there is a probability distribution μ over partial partitions of X, with the following properties.

- *Diameter bound:* Every partial partition $\widehat{P} \in \mathrm{supp}(\mu)$ is connected and Δ-bounded.
- *Separation event:* For all $u, v \in X$, $\Pr_{\widehat{P}\in\mu}[\exists S \in \widehat{P}$ such that $u \in S$ but $v \notin S]$ $\le \hat\beta \cdot d(u,v)/\Delta$.
- *Terminal partition:* For all $x \in K$, every partial partition $\widehat{P} \in \mathrm{supp}(\mu)$ has a cluster containing x.
- *Terminal-centered clusters:* For every partial partition $\widehat{P} \in \mathrm{supp}(\mu)$, every cluster $S \in \widehat{P}$ contains a terminal.

A graph $G = (V,E)$ with terminals K is $\hat\beta$-terminal decomposable if for every nonnegative length ℓ_G assigned to its edges, the resulting shortest-path metric d_G with terminals K is $\hat\beta$-terminal decomposable. Throughout, we assume that there is a polynomial-time algorithm that, given the metric, terminals, and Δ as input, samples a partial partition $\widehat{P} \in \mu$. Note that if $K = V$, the above definitions coincide with the definitions of β-decomposable metrics and graphs.

Our main theorem for terminal decomposable metrics is the following.

THEOREM 20. *Given $(G=(V,E), \ell_G)$, suppose d_G is $\hat\beta$-terminal decomposable with respect to terminals K. There is a randomized polynomial-time algorithm that produces a connected* 0*-extension $f : V \to K$ such that for all $u, v \in V$, we have $\mathbb{E}[d_G(f(u), f(v))] \le O(\hat\beta^2 \log\hat\beta) \cdot d_G(u,v)$.*

This theorem is interesting when $\hat\beta$ is much less than β, the decomposability of the metric itself; e.g., one can alter the CKR decomposition scheme to get $\hat\beta(k,n) = O(\log k)$, while $\beta = O(\log n)$.

4.2.1. The modified algorithm. Algorithm 2 for the terminal-decomposable case is very similar to Algorithm 1: the main difference is that in each iteration we only obtain a partial partition of the vertices; we map only the nodes that lie in clusters of this partial partition.

A few words about the algorithm: Recall that a partial partition returns a set of connected diameter-bounded clusters such that each cluster contains at least one terminal, and each terminal is in exactly one cluster—we use V^x to denote the cluster containing $x \in K$. (Hence either $V^x = V^y$ or $V^x \cap V^y = \emptyset$.) Now when we delete all the vertices in some cluster V^x that are already mapped, this includes the terminal x—and hence there is at least one candidate for w_C in line 9. Eventually, there will be only one cluster, in which case all vertices are mapped and the algorithm terminates.

The analysis for Theorem 20 is almost the same as for Theorem 12; the only difference is that Claim 19 is replaced by the following weaker claim which immediately gives the $O(\hat\beta^2 \log\hat\beta)$ bound.

Algorithm 2. Algorithm for connected 0-extension: The terminal-decomposable case.

input: $(G, \ell_G), K$.
let $i \leftarrow 0$, $f(x) = x$ for all $x \in K$, $f(v) = \perp$ for all $v \in V \setminus K$.
while there is a v such that $f(v) = \perp$ **do**
  let $i \leftarrow i + 1$, $r_i \leftarrow 2^i$
  find a $\hat{\beta}$-terminal decomposition of d_G with diameter bound r_i; let V^x be the cluster containing terminal x.
  for all clusters V^x in the partial partition **do**
    delete all vertices u in V^x with $f(u) \neq \perp$
    for each connected component C from V^x thus formed **do**
      choose a vertex $w_C \in V^x$ that was deleted and had a neighbor in C
      reset $f(u) = f(w_C)$ for all $u \in C$.
    end for
  end for
end while

Claim 21. $\Pr[\mathcal{B}_j] \leq \min\{8\hat{\beta}\frac{z}{2^j}, 1\} \cdot 23\hat{\beta}^2 \frac{d(u,v)}{2^j}$.

Proof. Recall that an edge is *untouched* after round j' if neither of its endpoints is mapped at the end of this round. For this to happen, u must be separated from its closest terminal in the clustering in round j', which happens with probability at most $\min\{\hat{\beta}\frac{A_e}{2^{j'}}, 1\}$. Also recall that the probability that an edge $e = (u, v)$ is cut in a round j' is at most $\hat{\beta}\frac{d(u,v)}{2^{j'}}$. Let i denote the round in which the edge is first touched. We upper bound the probability of the event $\mathcal{B}_j$ separately depending on how i and j compare. Note that for $j \leq 3$, the right-hand side is at least 1 so the claim holds trivially.

- $i \leq j - 3$. For $\mathcal{B}_j$ to occur, it must happen that the edge is cut in round i and it is either untouched or cut in rounds $j - 1$ and $j - 2$. The probability for this to happen is at most $\min\{\hat{\beta}\frac{d(u,v)}{2^i}, 1\} \cdot \min\{\hat{\beta}(\frac{A_e}{2^{j-2}} + \frac{d(u,v)}{2^{j-2}}), 1\} \cdot \hat{\beta}(\frac{A_e}{2^{j-1}} + \frac{d(u,v)}{2^{j-1}}) \leq \min\{\frac{d(u,v)}{2^i}, 1\} \min\{8\hat{\beta}\frac{z}{2^j}, 1\} \cdot 4\hat{\beta}^2\frac{z}{2^j}$. If $d(u, v) \geq A_e$ this is at most $\min\{8\hat{\beta}\frac{z}{2^j}, 1\} \cdot 16\hat{\beta}^2\frac{d(u,v)}{2^j}$ as $z = d(u, v)$. Otherwise, observe that $i \geq j_e$ as the edge cannot be touched before. Hence $2^i \geq A_e/4$, and plugging this in gives a bound of $\min\{8\hat{\beta}\frac{z}{2^j}, 1\} \cdot 16\hat{\beta}^2\frac{d(u,v)}{2^j}$, as well.
- $i = j - 2$. For $\mathcal{B}_j$ to occur, the edge e must be cut in round $j - 2$ and it must be cut or untouched in round $j - 1$, as otherwise it would already be settled in one of these rounds. The probability of this is at most $\hat{\beta}\frac{d(u,v)}{2^{j-2}} \cdot \min\{\hat{\beta}(\frac{d(u,v)}{2^{j-1}} + \frac{A_e}{2^{j-1}}), 1\} \leq \min\{4\hat{\beta}\frac{z}{2^j}, 1\} \cdot 4\hat{\beta}\frac{d(u,v)}{2^j}$.
- $i = j - 1$. For $\mathcal{B}_j$ to occur, the edge e must be cut in round $j - 1$ and must be untouched in round $j - 2$. The probability of this is at most $\min\{\hat{\beta}\frac{A_e}{2^{j-2}}, 1\} \cdot \hat{\beta}\frac{d(u,v)}{2^{j-1}} \leq \min\{4\hat{\beta}\frac{z}{2^j}, 1\} \cdot 2\hat{\beta}\frac{d(u,v)}{2^j}$.
- $i = j$. For $\mathcal{B}_j$ to occur, e must be cut in round j and must be untouched in round $j - 1$. The probability of this is at most $\min\{\hat{\beta}\frac{A_e}{2^{j-1}}, 1\} \cdot \hat{\beta}\frac{d(u,v)}{2^j} \leq \min\{4\hat{\beta}\frac{z}{2^j}, 1\} \cdot \hat{\beta}\frac{d(u,v)}{2^j}$.

Since $\Pr[\mathcal{B}_i] = \Pr[\mathcal{B}_i \wedge (i \leq j-3)] + \Pr[\mathcal{B}_i \wedge (i = j-2)] + \Pr[\mathcal{B}_i \wedge (i = j-1)] + \Pr[\mathcal{B}_i \wedge (i = j)]$, the claim follows. ∎

4.3. Connected 0-extension on general graphs. Finally, we show that for general metrics, we can do better than the $O(\log^2 k \log\log k)$ guarantee implied by Theorem 20. In particular, we now prove Theorem 15, which gives a $O(\log k)$ guarantee. We still use Algorithm 1 from the previous section, but use a specific decomposition algorithm. The following result follows from Fakcharoenpol et al. [12], who built upon the work of Calinescu, Karloff, and Rabani [5].

THEOREM 22 (see [12]). *Let $(G = (V, E), \ell_G)$ with a terminal set $K = \{x_1, \ldots, x_k\} \subseteq V$. There is a (randomized) polynomial-time algorithm that produces, for each $i = 0, 1, \ldots, \lceil \log \operatorname{diam}(G) \rceil$, a collection of $k+1$ clusters $\{C_0^i, C_1^i, \ldots, C_k^i\}$, such that*

(a) *(diameter) for any $j \neq 0$, C_j^i contains the terminal x_j, and $d(x_j, v) \leq 2^i$ for any $v \in C_j^i$;*

(b) *(separation) for any $u, v \in X$, $\Pr[\exists j$ such that $u \in C_j^i$ but $v \notin C_j^i] \leq O(\beta_i^{uv}) \cdot d(u,v)/2^i$, where the probability is taken over the internal coin tosses of the algorithm, and*

(c) *(amortization) for any $u, v \in X$, $\sum_i \beta_i^{uv} \leq \beta = O(\log k)$;*

(d) *(coverage) $\cup_{j\neq 0} C_j^i$ contains $\cup_{j=1}^k B_d(x_j, 2^{i-1})$.*

We remark that we do not need each cluster to induce a connected component. Observe that the (diameter) and (coverage) properties imply

(e) *(laminarity) for any i, $\cup_{j\neq 0} C_j^{i+1} \supseteq \cup_{j\neq 0} C_j^i$ with probability 1. Hence also $C_0^i \supseteq C_0^{i+1}$ with probability 1.*

We run Algorithm 1 with this decomposition; the only worry is that since the clusters are not connected, it may be the case that in step 9 we may not find a node w_C as desired. In this case, we *expel* C from C_s, and do not map the vertices in C in this iteration. This ensures the connectivity property of the $f_i^{-1}(x)$'s. Moreover, the laminarity property inductively ensures that we never map any vertex from C_0^i by the end of round i. Since the diameter property bounds the diameter of every other cluster, Lemma 17 continues to hold.

Now, by its very definition, any expulsion operation only removes components that are disconnected from the rest of C_s, and hence does not increase the separation probability for any edge. Moreover, it is still the case that if u is mapped before round j and an edge (u, v) is not cut in round j, then the node v gets mapped in round j as well. Indeed by laminarity, u is in one of the clusters containing a terminal, and if (u, v) is not cut, then v is in that cluster too. Since u is mapped, the component containing v cannot be expelled. Thus Claim 19 continues to hold and bounds the probability of $\mathcal{B}_j$, implying that

$$\begin{aligned}
\mathbb{E}[d(f(u), f(v))] &= \textstyle\sum_j \Pr[\mathcal{B}_j] \cdot \mathbb{E}[d(f(u), f(v)) \mid \mathcal{B}_j] \\
&\leq O(d_G(u,v)) + \textstyle\sum_{j\geq j'} \Pr[\mathcal{B}_j] \cdot O(2^j) \\
&\leq O(d_G(u,v)) + \textstyle\sum_{j\geq j'} \min\{4\beta \frac{z}{2^j}, 1\} \cdot 5\beta_i^{uv} \frac{d_G(u,v)}{2^j} \cdot O(2^j) \\
&\leq O(\beta\, d_G(u,v)).
\end{aligned}$$

Since $\beta = O(\log k)$, this gives us connected 0-extensions where the stretch is $O(\log k)$, and hence finishes the proof of Theorem 15.

5. Lower bounds.

In this section, we show two kinds of lower bounds. The first shows that any flow sparsifier that is a convex combination of 0-extensions must suffer a loss of $\Omega(\sqrt{\log k})$—for such an extension, this improves on the $\Omega(\log \log n)$ lower bound for (arbitrary) flow sparsifiers [26]. The second shows that any flow sparsifier that only uses edge capacities which are bounded from below by a constant, must suffer a loss of $\Omega(\sqrt{\log k}/\log\log k)$.

5.1. Lower bounds for 0-extension-based sparsifiers.

The following result can be viewed as following from the duality between 0-extensions and 0-extension-based flow sparsifiers (Theorem 5); by that theorem, not only do good 0-extension algorithms give good 0-extension-based flow sparsifiers, but the converse would also be true—and hence one can use a lower bound of Calinescu, Karloff, and Rabani [5] to infer lower bounds on 0-extension-based flow sparsifiers. The following theorem gives the explicit construction obtained thus.

THEOREM 23. *For infinitely many values of* k*, there is a graph* $G' = (V(G'), E(G'))$ *and a set* $K \subseteq V$ *of size* k *for which any flow sparsifier that is a convex combination of* 0*-extension graphs has quality at least* $\Omega(\sqrt{\log k})$.

Proof. We use the lower bound of $\Omega(\sqrt{\log k})$ on the 0-extension integrality ratio by Calinescu, Karloff, and Rabani [5]. For completeness we describe their construction: Let G be an expander with n vertices, maximum degree Δ, and expansion at least α, where Δ and α are fixed parameters. Define $l = \lceil \sqrt{\log n} \rceil$ and $k = \lceil \frac{n}{l} \rceil$. Choose any k distinct vertices $h_1, \ldots . h_k \in V(G)$ and add k new paths of length l starting at these vertices and ending at new vertices labeled $1, \ldots, k$. Denote the resulting graph by G' (note that $|V(G')| = O(n)$ and $|E(G')| = O(n)$), and let the terminals K be the new vertices $\{1, \ldots, k\}$. Set the costs and lengths of the edges to 1. The distance $d_{G'}(u, v)$ is set to be the shortest path distance in G' between u, v. For the described instance G', K of the 0-extension problem, Calinescu, Karloff, and Rabani show that

$$\sum_{e=(u,v)\in E(G')} c(e) d_{G'}(u, v) = |E(G')| = O(n),$$

while there exists a universal $\gamma > 0$ such that for any 0-extension function $f : V(G') \to K$,

$$\sum_{e=(u,v)\in E(G')} c(e) d_{G'}(f(u), f(v)) \geq \gamma n \sqrt{\log n} = \Omega\left(n\sqrt{\log k}\right).$$

We now use the instance G', K as follows. By [26, proof of Theorem 1] it is known that for any convex combination of 0-extensions $H = \sum \lambda_i H_i$, the quality of H is

$$\sup_{d_{G'} \text{ s.t. } \sum_e c(e) d_{G'}(e)=1} \left\{ \sum_{s,t\in K} c_H(s,t) d_{G'}(s,t) \right\}$$

$$= \sup_{d_{G'} \text{ s.t. } \sum_e c(e) d_{G'}(e)=1} \left\{ \sum_{f_i} \lambda_i \sum_{(u,v)\in E(G')} c(e) d_{G'}(f_i(u), f_i(v)) \right\}.$$

(The proof of this uses strong duality for the maximum concurrent flow problem.) We now show that there exists a semimetric $d_{G'}$ such that $\sum_e c(e) d_{G'}(e) = 1$, and for every 0-extension function $f : V(G') \to K$,

$$\sum_{(u,v)\in E(G')} c(e) d_{G'}(f(u), f(v)) = \Omega\left(\sqrt{\log k}\right). \tag{1}$$

We set $d_{G'}(e)$ to be $1/|E(G')|$ for every $e \in E(G')$. Thus, $\sum_{e\in E(G')} c(e) d_{G'}(e) = 1$. We set $d_{G'}(u, v)$ to be the shortest path distance between u, v in G' with respect to edge lengths $d_{G'}(e)$. From the above it follows that for every 0-extension function f,

$$\sum_{(u,v)\in E(G')} c(e) d_{G'}(f(u), f(v)) \geq \frac{\gamma n \sqrt{\log n}}{|E(G')|} = \Omega\left(\sqrt{\log n}\right) = \Omega\left(\sqrt{\log k}\right).$$

This proves (1), completing the proof. ∎

5.2. Lower bounds for sparsifiers having no small edges.

THEOREM 24. *For infinitely many values of k, there is a graph $G = (V, E)$ and a terminal set $K \subset V$ of size k for which any flow sparsifier with edge capacities at least $\varepsilon > 0$ has quality at least $\Omega(\varepsilon\sqrt{\log k}/\log\log k)$.*

Proof. Let n be a sufficiently large prime. Let $G = (V, E)$ be a graph whose nodes correspond to the elements of $\mathbb{Z}_n$ and that contains an edge $\{u, v\}$ if $v = u + 1$, $v = u - 1$, or $v = u^{-1}$ (all operations are w.r.t. $\mathbb{Z}_n$ and we define 0^{-1} as 0). In other words the graph consists of a Hamiltonian cycle plus some additional edges. This graph G is a 3-regular expander (see, e.g., [19]).

Choose the set of terminals K as $\{i \cdot \lceil\sqrt{\log n}\rceil \mid 0 \le i \le k-1\}$, with $k = n/\lceil\sqrt{\log n}\rceil$. To simplify notation, we will omit floor and ceiling operations in the following. For $i \in [0, k-1]$, let B_i be the set of the $\sqrt{\log n}$ nodes on the Hamiltonian cycle between terminal i and $i + 1$, including i but excluding $i + 1$.

Let $H = (K, E_H)$ be a flow sparsifier for G with edge capacities at least $\varepsilon > 0$. Let d be the maximum weight degree of H, where the weighted degree of a node is the sum over all capacities of incident edges.

CLAIM 25. *The maximum weighted degree d of H is at least*

$$c' \cdot \varepsilon \cdot \frac{\sqrt{\log n}}{\log\log n}$$

for some constant c'.

Proof. Consider a demand of $1/k$ between all pairs of terminals.

Since the minimum edge capacity is at least ε, the unweighted degree of H is at most d/ε. Due to this bounded degree, for sufficiently large k, there are at least $k^2/4$ terminal pairs that have distance at least $\log k/(2\log(d/\varepsilon))$ from each other (see, e.g., [5, Lemma 4.2]).

Each of these pairs induces a load of $1/k$ on at least $\log k/(2\log(d/\varepsilon))$ edges. Therefore, the total load in the network is at least $k\log k/(8\log(d/\varepsilon))$. Since H has at most $k \cdot d/(2\varepsilon)$ edges, the congestion in H is at least $\varepsilon\log k/(4d\log(d/\varepsilon))$.

The same demand can be routed with congestion at most $(c+1)\sqrt{\log n}$ in G, for some constant c depending on the edge expansion of G. Say each terminal i sends a total flow of 1. We can distribute this flow evenly between the nodes in B_i using only edges inside of B_i and with congestion of at most 1. This can easily be done, since we can send this flow along the Hamiltonian cycle to reach every node in B_i. Now, we route a uniform multicommodity flow on the whole expander, where the flow leaving each node is $1/\sqrt{\log n}$, i.e., the demand between every pair of nodes is $1/(n\sqrt{\log n})$. This requires congestion at most $c\log n \cdot (1/\sqrt{\log n}) = c\sqrt{\log n}$ [27]. Finally, the flow in each B_i is routed inside B_i to the respective terminal. Again, this can easily be done with congestion 1. In total, we sent a flow of $1/k$ between all pairs of terminals and the congestion is bounded by $c\sqrt{\log n} + 2 \le (c+1)\sqrt{\log n}$.

Hence, we identified a demand that requires congestion at least $\varepsilon\log k/(4d\log(d/\varepsilon))$ in H but can be routed with congestion at most $(c+1)\sqrt{\log n}$ in G. Since H is a flow sparsifier, its congestion has to be bounded by the congestion in G and, thus, $\varepsilon\log k/(4d\log(d/\varepsilon)) \le (c+1)\sqrt{\log n}$. It follows that

$$\frac{d}{\varepsilon}\log\left(\frac{d}{\varepsilon}\right) \ge \frac{\log k}{4(c+1)\sqrt{\log n}} \ .$$

Using the fact that $k = n/\sqrt{\log n}$, the claim follows. ∎

Now pick a node in H that has weighted degree at least $c' \cdot \varepsilon \cdot \sqrt{\log n}/\log\log n$ (such a nodes exists due to Claim 25). Consider the situation in which the demand between this node and every other node corresponds to the capacity of the edge connecting them in H, and all other demands are 0. Clearly, in H this can be routed with congestion 1. The terminal in G corresponding to node u, however, has only degree 3. Therefore, routing this demand in G results in congestion at least $c' \cdot \varepsilon \cdot \sqrt{\log n}/(3\log\log n) \geq c' \cdot \varepsilon \cdot \sqrt{\log k}/(3\log\log k)$, since that is the load on at least one of the outgoing edges of u. $\square$

6. Applications. Most of these applications were considered by Moitra [29], and Leighton and Moitra [26]; we show how our results above give improved approximations to the problems.

6.1. Steiner oblivious routing. Theorem 9 is an exact analogue of Räcke's theorem on general flows [31] for the special case of K-flows, and hence immediately gives an $O(\log k)$-oblivious routing scheme for K-flows.

6.2. Steiner minimum linear arrangement. Given $G = (V, E)$ and $K \subseteq V$ with $|K| = k$, the goal in the SMLA problem is to find a mapping $F : V \to [k]$ such that $F|_K : K \to [k]$ is a bijection. The goal is to minimize $\sum_{(u,v)\in E} c_{uv}|F(u)-F(v)|$. Note that for the non-Steiner minimum linear arrangement (MLA) case, where $K = V$, Rao and Richa [32] gave an $O(\log n)$-approximation for general graphs and an $O(\log\log n)$-approximation for graphs that admit $O(1)$-padded decompositions (which includes the family of all trees).

For our algorithm, we take a random tree/retraction pair (T, f) from the distribution of Theorem 3; this ensures that the cost of the optimal map F^* (viewed as a solution to the MLA problem on T) increases by an expected $O(\log k)$-factor. Now solving the MLA problem on the tree to within an $O(\log\log k)$ factor to get a map $\widehat{F}_T : K \to [k]$, and defining $\widehat{F}(x) = \widehat{F}_T(f(x))$ gives us an expected $O(\log k \log\log k)$-approximation. We show in section 7 that this can be improved slightly to $O(\log k)$ using a more direct approach.

6.3. Steiner graph bisection. In this problem, we are given a value k' and want to find a bipartition $(A, V \setminus A)$ of the graph such that $|A \cap K| = k'$, and that minimizes the cost of edges cut by the bipartition. We use Theorem 9 to embed the graph into a random tree losing an $O(\log k)$ factor. On this tree we use the approach of Räcke [31] to find the best $(k', k-k')$ bipartition on that. This gives us an $O(\log k)$ algorithm for this partitioning problem.

6.4. Steiner ℓ-multicut. In this problem, we are given terminal pairs $\{s_i, t_i\}_{i\in[k]}$, and a value $k' \leq k$, and we want to find a minimum cost set of edges whose deletion separates at least k' terminal pairs. Again, we can use Theorem 9 to embed the graph into a random tree losing an $O(\log k)$ factor, and use the theorem of Golovin, Nagarajan, and Singh [15] to get a $4/3 + \epsilon$-approximation on this tree; this gives us the randomized $O(\log k)$-approximation.

6.5. Steiner min cut linear arrangement. The Steiner min cut linear arrangement (SMCLA) problem is defined as follows: Given $G = (V, E)$ and $K \subseteq V$ with $|K| = k$, we want to find a mapping $F : V \to [k]$ such that $F|_K : K \to [k]$ is a bijection. The goal is to minimize $\max_i \sum_{x\in F^{-1}([i]), y\notin F^{-1}([i])} c_{xy}$. For the non-Steiner version of the problem, Leighton and Rao [27] show that given an α-approximation to the balanced partitioning (or to the bisection) problem, one can get an $O(\alpha \log n)$-

approximation to the min cut linear arrangement (MCLA) problem. Using [4], this gives an $O(\log^{1.5} n)$-approximation to the MCLA problem.

We note that the reduction works immediately for the Steiner version of the problem: given an α-approximation to Steiner bisection, one gets an $O(\alpha \log k)$-approximation to SMCLA. Thus we get an $O(\log^2 k)$-approximation to the SMCLA problem. We show in section 7 that this can be improved to $O(\log^{1.5} k)$ using a more direct approach.

7. Better algorithms using a direct approach.

The vertex sparsifiers give a modular approach to solving the Steiner version of various problems. Not surprisingly, for some of these problems, a direct attack will lead to better algorithms. In this section, we show that applying known techniques for the MLA problem leads to a better approximation ratio for SMLA, and for SMCLA.

7.1. Steiner minimum linear arrangement.

Recall that the SMLA problems is defined as follows: Given $G = (V, E)$ and $K \subseteq V$ with $|K| = k$, the goal is to find a mapping $F : V \to [k]$ such that $F|_K : K \to [k]$ is a bijection. The goal is to minimize $\sum_{(u,v)\in E} c_{uv}|F(u) - F(v)|$. Specifically, we show the following result.

THEOREM 26. *There is a polynomial time $O(\log k)$-approximation algorithm for the SMLA problem based on the natural LP relaxation.*

Proof. The linear program for the SMLA problem is based on the spreading metric LP relaxation for MLA introduced in [11]:

$$\begin{aligned}
\min \quad & \sum_{(u,v)\in E} c_{uv} d_{uv} & \\
\text{subject to} \quad & & \\
\text{(triangle inequality)} \quad & d_{uw} - d_{uv} - d_{vw} \le 0 & \forall u, v, w \in V, \\
\text{(spreading)} \quad & \sum_{v\in S} d_{uv} \ge \tfrac{|S|^2}{5} & \forall S \subseteq K, |S| \ge 2, u \in S, \\
& d_{uv} \ge 0 & \forall u, v \in V.
\end{aligned}$$

It follows from [11] that the above is a valid LP relaxation to the SMLA problem, and that one can efficiently separate for the spreading constraints so that the LP can be solved in polynomial time using the Ellipsoid algorithm. Further, it is easy to check that the spreading constraints imply that for any $u \in K$, $|\mathbf{B}_d(u, r) \cap K| \le 5r$. (Here, $\mathbf{B}_d(v, r) = \{w \mid d(v, w) \le r\}$ is the "ball" around v of radius r in the metric d.)

Let d be a solution to the above linear program. Since d is a metric on V, it follows from Theorem 3 that we construct a (random) edge-weighted 2-HST $T = (I \cup K, E_T)$ with internal nodes I and leaves K, and a retraction $f : V \to K$ such that

(a) $d_T(f(x), f(y)) \ge d(x, y)$ for all $x, y \in K$ (with probability 1),

(b) $\mathbb{E}_T[d_T(f(u), f(v))] \le O(\log k)\, d(u, v)$ for all $u, v \in V$.

We argue that given this HST, we can construct a mapping $F_T : V \to [k]$ such that $F_T|_K : K \to [k]$ is a bijection. This mapping will have the property that $|F_T(u) - F_T(v)| \le 5 d_T(f(u), f(v))$. The approximation ratio of $O(\log k)$ then follows from property (b) above.

The mapping F_T is defined by taking the natural left-to-right ordering on K defined by T, and assigning every other vertex $v \in V$ to the position $f(v)$. Formally, let π be a preorder traversal of T. For every terminal $x \in K$, set $F_T(x)$ to the number of terminals in π that occur before x, i.e., $F_T(x) = |K \cap \{\pi_i : i \le \pi^{-1}(x)\}|$. For every other vertex $u \in V$, set $F_T(u) = F_T(f(u))$. It is easy to check that $F_T|_K$ is a bijection.

We next upper bound $|F_T(u) - F_T(v)|$ for $u, v \in V$. Consider the terminals $t_u = f(u), t_v = f(v)$: If $t_u = t_v$, then $F_T(x) = F_T(y)$ and there is nothing to prove; else let T_{uv} be the smallest subtree of T containing t_u and t_v. By the properties of the HST, we have $d_T(t_u, t_v) \geq d_T(t_u, z)$ for all $z \in T_{xy}$. Moreover, $d_T(u, v) = d_T(t_u, t_v)$. Now,

$$\begin{aligned}
& & |F_T(u) - F_T(v)| &= |F_T(t_u) - F_T(t_v)| \\
& & &\leq |K \cap T_{uv}| \\
&\text{(since } d_T(t_u, t_v) \geq d_T(t_u, z) \text{ for all } z \in T_{uv}) & &\leq |K \cap \mathbf{B}_{d_T}(t_u, d_T(t_u, t_v))| \\
&\text{(by property (a))} & &\leq |K \cap \mathbf{B}_{d}(t_u, d_T(t_u, t_v))| \\
&\text{(by the spreading property)} & &\leq 5 d_T(t_u, t_v) \\
& & &= 5 d_T(u, v).
\end{aligned}$$

This proves Theorem 26. ∎

7.2. Steiner min cut linear arrangement. Recall that the SMCLA problem is defined as follows. Given $G = (V, E)$ and $K \subseteq V$ with $|K| = k$, the goal is to find a mapping $F : V \to [k]$ such that $F|_K : K \to [k]$ is a bijection. The goal is to minimize $\max_i \sum_{x \in F^{-1}([i]), y \notin F^{-1}([i])} c_{xy}$. Specifically, we show the following result.

THEOREM 27. *There is a polynomial-time $O(\log^{1.5} k)$-approximation algorithm for the SMCLA problem.*

The algorithm and the proof are the natural generalization of the $O(\log^{1.5} n)$ approximation to the MCLA problem. We sketch the argument here.

This algorithm is based on an SDP formulation and the sparsest cut algorithm of [4], where the following theorem is shown.

THEOREM 28. *There exists a constant $\varepsilon > 0$ such that the following holds. For any k-point ℓ_2^2 metric (S, d) satisfying $\sum_{x,y \in S} d_{xy} \geq \frac{|S|^2}{8}$, there are sets $A, B \subseteq S$ such that $|A|, |B| \geq \varepsilon k$ and $d(A, B) \geq \frac{\varepsilon}{\sqrt{\log k}}$. Moreover given vectors $\{v_x : x \in S\}$ representing d, such sets A, B can be found in polynomial time.*

Consider first the following linear program:

$$\begin{aligned}
\min \quad & \sum_{(x,y) \in E} c_{xy} d_{xy} \\
\text{subject to} \quad & \\
\text{(triangle inequality)} \quad & d_{xz} - d_{xy} - d_{yz} \leq 0 \qquad \forall x, y, z \in V, \\
\text{(balance)} \quad & \sum_{x,y \in K} d_{xy} \geq \frac{|K|^2}{8}, \\
& d_{xy} \geq 0 \qquad \forall x, y \in V.
\end{aligned}$$

Let $F : V \to [k]$ be the optimum MCSLA with value OPT. Then the cut separating $F^{-1}([\lfloor \frac{k}{2} \rfloor])$ from its complement has value at most OPT, and gives a feasible integral solution to above linear program. Thus the value of the relaxation above is at most OPT.

Suppose in the above linear program, we additionally require that the distance metric d be an ℓ_2^2 metric, i.e., there exists vectors $v_x \in \mathbb{R}^n$ such that $d(x, y) = \|v_x - v_y\|_2^2$. This program can be naturally written as an SDP, and can be solved in polynomial time to return vectors $\{v_x\}$. Moreover, the optimum to this relaxation has value at most OPT as well. Theorem 28 then implies that we can find sets

$A, B \subseteq K$ such that $|A|, |B| \geq \varepsilon k$ and where $d(A,B) \geq \Delta = \frac{\varepsilon}{\sqrt{\log k}}$. Consider the sets $A_r = \{x \in V : d(A,x) \leq r\}$. For $0 < r < \Delta$, it is immediate that $A \subseteq A_r \subseteq V \setminus B$.

Picking r at random from $(0, \Delta)$, we observe that for any $x, y \in V$

$$\Pr[x \in A_r, y \notin A_r] \leq (d(y,A) - d(x,A))/\Delta,$$

so that by the triangle inequality, the expected cost of the cut $(A_r, V \setminus A_r)$ is at most $\frac{1}{\Delta}\sum_{(x,y)\in E} c_{xy} d_{xy} \leq OPT/\Delta$. Thus we can find an $r \in (0, \Delta)$ such that

(a) $|K \cap A_r|, |K \cap (V \setminus A_r)| \leq (1-\varepsilon)k$;

(b) $\sum_{x \in A_r, y \notin A_r} c_{xy} \leq O(OPT\sqrt{\log k})$.

We can recursively compute Steiner linear arrangements for A_r and $V \setminus A_r$, and by condition (a), the depth of the recursion is at most $O(\log k)$. For any i, we can thus bound the total cost of edges from $F^{-1}([i])$ to $V \setminus F^{-1}([i])$. Indeed each level of the recursion contributes at most $O(OPT\sqrt{\log k})$ to this cost. Since there are at most $O(\log k)$ levels, we get an $O(\log^{1.5} k)$ approximation.

REFERENCES

[1] R. Andersen and U. Feige, *Interchanging distance and capacity in probabilistic mappings*, preprint, arXiv:0907.3631[cs.DS], 2009.

[2] A. Andoni, A. Gupta, and R. Krauthgamer, *Towards* $(1+\epsilon)$*-approximate flow sparsifiers*, in Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2014, pp. 279–293.

[3] A. Archer, J. Fakcharoenphol, C. Harrelson, R. Krauthgamer, K. Talwar, and É. Tardos, *Approximate classification via earthmover metrics*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM, New York, 2004, pp. 1079–1087.

[4] S. Arora, S. Rao, and U. V. Vazirani, *Expander flows, geometric embeddings and graph partitioning*, J. ACM, 56 (2009), 5.

[5] G. Calinescu, H. J. Karloff, and Y. Rabani, *Approximation algorithms for the* 0*-extension problem*, SIAM J. Comput., 34 (2004), pp. 358–372.

[6] T.-H. Hubert Chan, D. Xia, G. Konjevod, and A. Richa, *A tight lower bound for the Steiner point removal problem on trees*, in Approximation, Randomization, and Combinatorial Optimization, Lecture Notes in Comput. Sci. 4110, Springer, Berlin, 2006, pp. 70–81.

[7] M. Charikar, T. Leighton, S. Li, and A. Moitra, *Vertex sparsifiers and absract rounding algorithms*, in Proceedings of the 51st IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, Los Alamitos, CA, 2010, pp. 265–274.

[8] C. Chekuri, F. B. Shepherd, and C. Weibel, *Flow-cut gaps for integer and fractional multiflows*, J. Combin. Theory Ser. B, 103 (2013), pp. 248–273.

[9] J. Chuzhoy, *On vertex sparsifiers with Steiner nodes*, in Proceedings of the 44th ACM Symposium on Theory of Computing (STOC), ACM, New York, 2012, pp. 673–688.

[10] J. Chuzhoy, Y. Makarychev, A. Vijayaraghavan, and Y. Zhou, *Approximation algorithms and hardness of the k-route cut problem*, in Proceedings of the 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM, New York, 2012, pp. 780–799.

[11] G. Even, J. Naor, S. Rao, and B. Schieber, *Divide-and-conquer approximation algorithms via spreading metrics*, J. ACM, 47 (2000), pp. 585–616.

[12] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar, *An improved approximation algorithm for the* 0*-extension problem*, in Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, ACM, New York, 2003, pp. 257–265.

[13] J. Fakcharoenphol, S. Rao, and K. Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, J. Comput. System Sci., 69 (2004), pp. 485–497.

[14] J. Fakcharoenphol and K. Talwar, *Improved decompositions of graphs with forbidden minors*, in Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization, Springer-Verlag, Berlin, 2003, pp. 36–46.

[15] D. Golovin, V. Nagarajan, and M. Singh, *Approximating the* K*-multicut problem*, in Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, 2006, pp. 621–630.

[16] A. Gupta, *Steiner points in tree metrics don't (really) help*, in Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM, New York, 2001, pp. 220–227.

[17] A. Gupta, V. Nagarajan, and R. Ravi, *Improved approximation algorithms for requirement cut*, Oper. Res. Lett., 38 (2010), pp. 322–325.

[18] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair, *Cuts, trees and ℓ_1-embeddings of graphs*, Combinatorica, 24 (2004), pp. 233–269.

[19] S. Hoory, N. Linial, and A. Wigderson, *Expander graphs and their applications*, Bull. Amer. Math. Soc. (N.S.), 43 (2006), pp. 439–561.

[20] L. Kamma, R. Krauthgamer, and H. Nguyen, *Cutting corners cheaply, or how to remove Steiner points*, in Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2014, pp. 1029–1040.

[21] R. Khandekar, *Lagrangian Relaxation Based Algorithms for Convex Programming Problems*, Ph.D. thesis, Indian Institute of Technology Delhi, New Delhi, 2004.

[22] P. Klein, S. A. Plotkin, and S. B. Rao, *Excluded minors, network decomposition, and multicommodity flow*, in Proceedings of the 25th ACM Symposium on Theory of Computing (STOC), ACM, New York, 1993, pp. 682–690.

[23] J. R. Lee, M. Mendel, and M. Moharrami, *A node-capacitated Okamura-Seymour theorem*, in Proceedings of the 45th ACM Symposium on Theory of Computing (STOC), ACM, New York, 2013, pp. 495–504.

[24] J. R. Lee and A. Naor, *Extending Lipschitz functions via random metric partitions*, Invent. Math., 160 (2005), pp. 59–95.

[25] J. R. Lee and A. Sidiropoulos, *On the geometry of graphs with a forbidden minor*, in Proceedings of the 41st ACM Symposium on Theory of Computing (STOC), ACM, New York, 2009, pp. 245–254.

[26] T. Leighton and A. Moitra, *Extensions and limits to vertex sparsification*, in Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC), ACM, New York, 2010, pp. 47–56.

[27] T. Leighton and S. B. Rao, *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*, J. ACM, 46 (1999), pp. 787–832.

[28] K. Makarychev and Y. Makarychev, *Metric extension operators, vertex sparsifiers and Lipschitz extendability*, in Proceedings of the 51st IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, Los Alamitos, CA, 2010, pp. 255–264.

[29] A. Moitra, *Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size*, in Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, Los Alamitos, CA, 2009, pp. 3–12.

[30] S. Nowozin and C. H. Lampert, *Global connectivity potentials for random field models*, in Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Piscataway, NJ, 2009, pp. 818–825.

[31] H. Räcke, *Optimal hierarchical decompositions for congestion minimization in networks*, in Proceedings of the 40th ACM Symposium on Theory of Computing (STOC), ACM, New York, 2008, pp. 255–264.

[32] S. Rao and A. W. Richa, *New approximation techniques for some ordering problems*, in Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM, New York, 1998, pp. 211–218.

[33] S. Vicente, V. Kolmogorov, and C. Rother, *Graph cut based image segmentation with connectivity priors*, in Proceedings of the 21st IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Piscataway, NJ, 2008.