Vertex Sparsifiers: New Results from Old Techniques[⋆]

Matthias Englert[1 ⋆⋆], Anupam Gupta[2 ⋆⋆⋆], Robert Krauthgamer[3 †], Harald Räcke[1 ‡], Inbal Talgam-Cohen[3], and Kunal Talwar[4]

[1] Department of Computer Science and DIMAP, University of Warwick, Coventry, UK.
[2] Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA.
[3] Weizmann Institute of Science, Rehovot, Israel.
[4] Microsoft Research Silicon Valley. Mountain View, CA, USA.

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 [FOCS 2009] and Leighton and Moitra [STOC 2010], 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.

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 [Räc08] give a convex combination of

[⋆] A full version appears at `http://arxiv.org/abs/1006.4586`

[⋆⋆] Supported by EPSRC grant EP/F043333/1 and DIMAP (the Centre for Discrete Mathematics and its Applications).

[⋆⋆⋆] Research was partly supported by the NSF award CCF-0729022, and an Alfred P. Sloan Fellowship. Research done when visiting Microsoft Research SVC.

[†] Supported in part by The Israel Science Foundation (grant #452/08), and by a Minerva grant.

[‡] Supported by DIMAP (the Centre for Discrete Mathematics and its Applications).

trees H with a loss of $O(\log n)$. We call this a *tree-based flow-sparsifier*—it uses a convex combination of trees.[5] 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 [Moi09] and by Leighton and Moitra [LM10] 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 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 poly-logarithmically 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 [Räc08] to H', 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 [Räc08] (which was reinterpreted in an abstract setting by Andersen and Feige [AF09]), we obtain the following improvements over the results of [Moi09, LM10].[6]

1. We show that using the 0-extension results [CKR04, FHRT03] in the framework of [Räc08, AF09] almost immediately gives us *efficent* constructions of flow-sparsifiers with loss $O(\frac{\log k}{\log \log k})$. While the existential result of [LM10] 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.
2. We then use a randomized tree-embedding due to [GNR10], which is a variant of the so-called FRT tree-embedding [FRT04] where the expected stretch is reduced to $O(\log k)$ by requiring the non-contraction condition only for terminal pairs. Using this refined embedding in the framework of [Räc08, AF09], we obtain 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 $\max_{u,v \in V} {}^{\mathbb{E}[d_{H_f}(f(u), f(v))]}/_{d_G(u,v)} \leq O(1)$. In particular, the planar graphs H_f produced are graph-theoretic minors of G. We remark that the known 0-extension algorithms [CKR04, AFH+04, LN05] do not ensure planarity of H_f.
It follows that planar graphs admit a *planar-based flow-sparsifier* (i.e., which 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 [LM10], but their sparsifiers are not planar-based.

[5] Given 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 $\mathcal{F}$.

[6] Recently, it has come to our attention that, independent of and concurrent to our work, Charikar, Leighton, Li, and Moitra, and independently Makarychev and Makarychev, obtained results similar to the first two below, as well as related lower bounds.

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 [Gup01]. We believe this randomized procedure is of independent interest; e.g., combined with an embedding of [GNRS04], this gives an alternate proof of the fact that the metric induced on the vertices of a single face of a planar graph can be embedded into a distribution over trees [LS09].

4. The results for planar graphs are in fact much more general. Suppose G is a β_G-decomposable graph (see definition in Section 1.1). 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 $\max_{u,v\in V} \mathbb{E}[d_{H_f}(f(u), f(v))]/d_G(u,v)$ is bounded by $O(\beta_G \log \beta_G)$. Now applying the same ideas of interchanging distance and capacity preservation, given any G and K, we can find *minor-based flow sparsifiers* with loss $O(\beta_G \log \beta_G)$.
5. Finally, we show 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 [LM10]. 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. 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 $k = n$.

	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	$O(\log n)^\dagger$, $O(\frac{\log^3 k}{\log\log k})$	$O(\log k)$	$O(\log n)$

Table 1. Summary of our results. Previous results marked with † from [Räc08], all others from [Moi09, LM10].

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 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 the full version we show how to get better approximations for the Steiner linear arrangement problems above using direct LP/SDP approaches.

1.1 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 with terminals $K \subset X$. 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) with terminals K is called *β-decomposable* if for every $\Delta > 0$ there is polynomial time algorithm to sample from a probability distribution μ over partitions of X, with the following properties:

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

β-decompositions of metrics have become standard tools with many applications; for more information see, e.g., [LN05].

We say that *a graph* $G = (V, E)$ is β-decomposable if for every nonnegative edge-lengths ℓ_G, the resulting shortest-path metric d_G is β-decomposable. Additionally, we assume that each cluster S in any partition P 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; the separation probability for non-adjacent pairs (u, v) can be bounded by $\beta \cdot d(u, v)/\Delta$ by noting that some edge on the u-v shortest path must be separated for (u, v) to be separated, and applying the union bound.

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. We also derive in Corollary 1 a variation of a known result on tree embeddings, which will be applied in Section 3.

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

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

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

Theorem 1 **([FHRT03]).** *There is an algorithm $\mathcal{A}_{FHRT}$ for 0-extension with stretch $\alpha = \alpha_{FHRT} := O(\frac{\log k}{\log\log k})$.*

Theorem 2 **([CKR04], see also [LN05]).** *If the graph is β-decomposable, there is an algorithm $\mathcal{A}_{CKR}$ for 0-extensions with stretch $\alpha = \alpha_{CKR} := O(\beta)$.*

In particular, if the graph G belongs to a non-trivial family of graphs that is minor-closed, it follows from [KPR93, FT03] that $\alpha = O(1)$.

2.1 0-Extension With Trees

The following result is a direct corollary of [GNR10, Theorem 7] (which in turn is an extension of the tree-embedding theorem of [FRT04]). Details omitted from this version.

Corollary 1 **(Tree 0-extension).** *There is a randomized polynomial-time algorithm $\mathcal{A}_{GNR}$ for 0-extension that has $\alpha_{GNR} = O(\log k)$; furthermore, the graphs output by the algorithm are trees on the vertex set K.*

As an aside, a weaker version of Corollary 1 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 $\mathbb{E}_H[d_H(x,y)] \leq O(\frac{\log k}{\log\log k})\ d_G(x,y)$ for all $x,y \in K$. Then use the result of [FRT04] to get a random tree $H' = (K, E_{H'})$ such that $\mathbb{E}_{H'}[d_{H'}(x,y)] \leq O(\log k)\ d_H(x,y)$ for all $x,y \in V(H)$. Combining these two results proves the weaker claim.

3 Flow-Sparsifiers

Recall that given an edge-capacitated graph (G,c) and a set $K \subseteq V$ of terminals, a *flow-sparsifier with quality* ρ 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

We use the framework of Räcke [Räc08], as interpreted by Andersen and Feige [AF09]. 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 $\lambda_M \geq 0$ for each $M \in \mathcal{M}$ such that $\sum_{M\in\mathcal{M}} \lambda_M = 1$.

Distance Mappings. Given $G = (V, E)$ and 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 $\{\lambda\}$ is $\sum_M \lambda_M(\sum_{e'} \mathbf{M}_{e,e'} \frac{\ell(e')}{\ell(e)})$.
- The *stretch of a probabilistic mapping* is the maximum over all edges of their average stretch.

Capacity Mappings. Given a graph G 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 $\{\lambda\}$ is $\sum_M \lambda_M(\sum_e \mathbf{M}_{e,e'} \frac{c(e)}{c(e')})$.
- The *congestion of a probabilistic mapping* is the maximum over all edges of their expected loads.

The Transfer Theorem. Andersen and Feige [AF09] distilled ideas from Räcke [Räc08] to state:

Theorem 3 (Theorem 6 in [AF09]). *Fix a graph G and a collection $\mathcal{M}$ of admissible mappings. For every $\rho \geq 1$, the following are equivalent:*

1. *For every collection of edge lengths ℓ_e, there is a probabilistic mapping with stretch at most ρ.*
2. *For every collection of edge capacities c_e, there is a probabilistic mapping with congestion at most ρ.*

In our settings, the techniques of Räcke [Räc08] can be used to make the result algorithmic: if one can efficiently sample from the probabilistic mapping with stretch ρ (which is true for the settings in this paper), one can efficiently sample from a probabilistic mapping with congestion $O(\rho)$ (and *vice versa*). In fact, one can obtain an explicit distribution on polynomially many admissible mappings. We defer further discussion of efficiency issues to the full version of the paper.

3.2 Tree-Based Flow Sparsifiers

The distance mappings we will consider will be similar to Räcke's application. Let us first fix for each $u, v \in K$ a canonical shortest-path S_{uv} between u, v in G. Now, consider a tree 0-extension (T, f) where $T = (K, E_T)$ and $f : V \to K$ is a retraction. For each edge $e = (w, x) \in E(G)$, consider the (unique) $f(w)$-$f(x)$-path $P_T(f(w), f(x))$ in the tree T. Define the mapping $M_T : E \to \mathcal{P}$ corresponding to the 0-extension (T, f) by

$$M_T((w, x)) = \uplus_{(u,v)\in P_T(f(w),f(x))} S_{uv}. \tag{3.1}$$

In other words, this maps each tree edge (w, x) to its canonical path; for each non-tree edge (w, x), it considers the edges on the tree-path between the images of w and x in the tree, and maps (w, x) to the disjoint union of the canonical paths of these edges. Recall that $M_T((w, x))$ is a multiset. In the corresponding matrix representation, $\mathbf{M}_{e,e'}$ is the multiplicity of e' in the set $\uplus_{(u,v)\in P_T(f(w),f(x))} S_{uv}$. Corollary 1 now implies the following:

Theorem 4. *Given a graph (G, ℓ) with terminals $K \subseteq V(G)$, there is a polynomial-time procedure to sample from a probabilistic mapping (which is a distribution over tree 0-extensions) with stretch $\rho_{dist} = O(\log k)$. Moreover, $\rho_{dist} \geq 1$ if $K \neq \emptyset$.*

Now we can apply the Transfer Theorem. Recall that in a K-flow, all source-sink pairs belong to set K.

Theorem 5 (Tree-Based Flow-Sparsifiers). *Given an edge-capacitated graph (G, c), and a set of terminals $K \subseteq V$, there is a polynomial-time algorithm that outputs a graph $H = (K, E_H)$ that is a convex combination of edge capacitated trees such that:*

(a) every K-flow that can be routed in G, can also be routed in H; and
(b) every K-flow that can be feasibly routed in H, can be routed with congestion $O(\log k)$ in G.

In other words, if we were to scale up the capacities in G to route all feasible flows in H, then the factor by which we would have to scale up capacities would only be $O(\log k)$.

Proof. We apply Theorem 3 and Theorem 4 to $G = (V, E)$ to get a convex combination $\{\lambda_{T,f}\}$ of maps $(T = (K, E_T), f)$ such that each edge in E has an average load of $O(\rho_{dist})$. Let us see how this implies (a) and (b) above: this is essentially a matter of unraveling the definitions. For each such (T, f), we define capacities on the edges $e_T \in E_T$ thus: let (A, B) be node sets of the two connected components of T formed by deleting the edge e_T, where $A \cup B = K$. Let $A' = \{v \in V \mid f(v) \in A\}$, and $B' = V \setminus A'$. Define

$$c_{T,f}(e_T) := \sum_{e\in E\cap(A'\times B')} c(e). \tag{3.2}$$

We claim that this convex combination $\{\lambda_{T,f}\}$ of capacitated trees satisfies (a) and (b). For (a), the definition of the capacities $c_{T,f}$ ensures that each edge of G can be concurrently routed feasibly in each T using capacities $c_{T,f}(\cdot)$, hence so can any K-flow feasible in G. Since this holds for each (T, f) pair, it holds for the convex combination.

To prove (b), we want to route edges in the convex combination of trees in the graph G, where we scale the capacities $c_{T,f}$ of edges from (T, f) by its convex multiplier $\lambda_{T,f}$. Consider any edge $e_T = (u, v) \in E_T$ with capacity $c_{T,f}(e_T)$ defined in (3.2): we can use the canonical shortest path S_{uv} to route this flow. Hence the load on any edge $e' = (w', x') \in E$ due to the convex combination of trees is at most

$$\frac{1}{c(e')} \sum_{T,f} \lambda_{T,f} \sum_{e_T\in E_T : e'\in S_{uv}} c_{T,f}(e_T). \tag{3.3}$$

Since $c_{T,f}(e_T)$ is the sum of the capacity of all edges $e = (w, x)$ such that e_T lies on the unique tree-path between $f(w), f(x)$, we rewrite (3.3) as

$$\frac{1}{c(e')} \sum_{T,f} \lambda_{T,f} \sum_{e_T=(u,v)\in E_T : e' \in S_{uv}} \sum_{(w,x)\in E : e_T \in P_T(f(w),f(x))} c(wx) \tag{3.4}$$

$$= \frac{1}{c(e')} \sum_{T,f} \lambda_{T,f} \sum_{(w,x)\in E} c(wx) \times (\text{multiplicity of } e' \text{ in } \uplus_{(u,v)\in P_T(f(w),f(x))} S_{uv}). \tag{3.5}$$

However, this is exactly the *expected load* for e' under the notion of admissible maps defined in (3.1); hence this is bounded by the congestion (the maximum expected load over all edges), which is at most ρ_{dist} by Theorem 3. This proves condition (b) above, that the congestion to route any K-flow in the convex combination H in the graph G is at most ρ_{dist}. □

3.3 General Flow Sparsifiers

Theorem 6 (Flow-Sparsifiers). *Given any graph G and terminals K, there is a randomized polynomial-time algorithm to output a flow-sparsifier H with loss $O(\frac{\log k}{\log \log k})$.*

Proof. Suppose we use Theorem 1 instead of using the tree 0-extension result (Corollary 1), we use the constructive version of the Transfer Theorem to get a polynomial number of graphs $H_1, H_2, \ldots$ on the vertex set K such that a convex combination of these graphs is a flow-sparsifier for the original graph G where the load is $O(\frac{\log k}{\log \log k})$. We can then construct a single graph H by setting the capacity of an edge to be the appropriate weighted combination of capacities of those edges in H_i; all feasible K-flows in G can be routed in H, and all feasible K-flows in H can be routed in G with congestion $O(\frac{\log k}{\log \log k})$. □

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

Theorem 7 (Flow-Sparsifiers for Minor-Closed Families). *For any graph G that is β-decomposable and any K, there is a randomized polynomial-time algorithm to construct a flow-sparsifier with loss $O(\beta)$.*

Note that the decomposability holds if G belongs to a non-trivial minor-closed-family $\mathcal{G}$ (e.g., if G is planar). However, Theorem 7 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 [KPR93, FT03]. Thus, Theorem 8, Corollary 5 and Theorem 9 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 8 (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 3 to interchange the distance preservation with capacity preservation, we get the following analogue of Theorem 5.

Corollary 2 (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 poly-time.*

Since planar graphs are $O(1)$-decomposable and since their minors are planar, by Corollary 2 they have an efficiently constructable planar-based flow-sparsifier with quality $O(1)$. By Theorem 8, 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 non-terminals and get a related planar graph on K while preserving inter-terminal distances in expectation. This generalizes a result of Gupta [Gup01] who showed a similar result for trees. (Obviously, this extends to every family of graphs excluding a fixed minor.)

Theorem 9 (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 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 $O(\log k)$; details in the full version.

Theorem 10 (Connected 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 stretch factor $\mathbb{E}[d_H(f(u), f(v))] \leq O(\log k) \cdot d_G(u, v)$ for all $u, v \in V$.*

Using the semi-metric 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 [VKR08, NL09].

4.1 The Algorithm for Decomposable Metrics

We now give the algorithm behind Theorem 8. Assume that edge lengths ℓ_G are integral and scaled such that 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

```
1:  input: $(G, \ell_G), K$.
2:  let $i \leftarrow 0$, $f(x) = x$ for all $x \in K$, $f(v) = \perp$ for all $v \in V \setminus K$.
3:  while there is a $v$ such that $f(v) = \perp$ do
4:      let $i \leftarrow i + 1$, $r_i \leftarrow 2^i$
5:      sample a $\beta$-decomposition of $d_G$ with diameter bound $r_i$ to get a partition $P$
6:      for all clusters $C_s$ in the partition $P$ that contains both mapped and unmapped vertices do
7:          delete all vertices $u$ in $C_s$ with $f(u) \neq \perp$
8:          for each connected component $C$ from $C_s$ do
9:              choose a vertex $w_C \in C_s$ that was deleted and had an edge to $C$
10:             reset $f(u) = f(w_C)$ for all $u \in C$.
11:         end for
12:     end for
13: 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 colored x. The following claim follows inductively:

Lemma 1. *For every 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 2. *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 $d(u, w_C) \le r_i$. Inductively, $d(w_C, x) \le 2r_{i-1}$ and the claim follows by triangle inequality. □

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

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

Note that for $e = (u, v)$, $d_G((f(u), f(v)) = d_H((f(u), f(v))$, and so it's enough to prove the claim for d_G. The analogous claim for non-adjacent 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 later 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 3. *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 end points of e is mapped before round j_e, then $2 \cdot 2^{j_e} \le A_e = d_G(e, K)$, which contradicts Lemma 2. For (b), both $d_G(u, f(u)), d_G(v, f(v)) \le 2^{j+1}$ by Lemma 2; the triangle inequality completes the proof. □

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

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

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

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

Lemma 3(b) implies that if the edge is settled before round $j_d := \lfloor \log(d_G(u,v)) \rfloor$, the conditional expectation $\mathbf{E}[d_G(f(u), f(v)) \mid \mathcal{B}_j]$ 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 3(a). Let $j_m := \max\{j_d, j_e\}$. It therefore suffices to to show that

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

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

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

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 8.

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 definition of decompositions in Section 1.1). 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 probability distribution μ over partial partitions of X, with the following properties:

- *Diameter bound:* Every partial partition $\widehat{P} \in \operatorname{supp}(\mu)$ 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 $\widehat{P} \in \operatorname{supp}(\mu)$ has a cluster containing x.
- *Terminal-centered clusters:* For every partial partition $\widehat{P} \in \operatorname{supp}(\mu)$, 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 lengths 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 11. *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 $\mathbb{E}[d_G(f(u), f(v))] \leq O(\hat{\beta}^2 \log \hat{\beta}) \cdot d_G(u, v)$.*

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 $\beta = O(\log n)$.

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 color 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.

Algorithm 2 Algorithm for Connected 0-extension: the terminal-decomposable case

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

The analysis for Theorem 11 is almost the same as for Theorem 8; the only difference is that Claim 4.1 is replaced by the following weaker claim (proof omitted from this version), which immediately gives the $O(\hat{\beta}^2 \log \hat{\beta})$ bound.

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

5 Future Directions

We gave a set of results on and around the idea of flow-sparsifiers and 0-extensions. Some of these results are not tight, and it would be interesting to obtain better bounds for these problems. Another interesting direction for future work is this: define an ℓ-*sparse-extension* of graph $G = (V, E)$ with terminals K to be any graph $H = (Z, E_H)$ with $|Z| = \ell$, $K \subseteq Z \subseteq V$, along with a retraction $f : V \to Z$ that satisfies $d_H(x, y) \geq d_G(x, y)$ for all $x, y \in Z$. (Note that a $|K|$-sparse-extension is just a 0-extension; one possible $|V|$-sparse-extension is G itself.) What if we consider ℓ-sparse-extensions (H, f) with

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

where ideally $\ell = \text{poly}(k)$, and $\alpha = O(1)$ (or just $\alpha \ll \frac{\log k}{\log \log k}$)? In other words, if we are willing to retain a small number of non-terminals, can we achieve better stretch bounds? Note that standard lower bounds for 0-extension have the property that $|V| = \text{poly}(k)$—hence the entire graph G is a "good" solution (poly(k)-sparse-extension with $\alpha = 1$).

References

[AF09] R. Andersen and U. Feige. Interchanging distance and capacity in probabilistic mappings. *CoRR*, abs/0907.3631, 2009.

[AFH+04] A. Archer, J. Fakcharoenphol, C. Harrelson, R. Krauthgamer, K. Talwar, and É. Tardos. Approximate classification via earthmover metrics. In *Proc. 15th SODA*, pages 1079–1087, 2004.

[CKR04] G. Calinescu, H. J. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. *SIAM J. Comput.*, 34(2):358–372, 2004.

[FHRT03] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar. An improved approximation algorithm for the 0-extension problem. In *Proc. 14th SODA*, pages 257–265, 2003.

[FRT04] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. System Sci.*, 69(3):485–497, 2004.

[FT03] J. Fakcharoenphol and K. Talwar. Improved decompositions of graphs with forbidden minors. In *Proc. 6th APPROX*, pages 36–46, 2003.

[GNR10] A. Gupta, V. Nagarajan, and R. Ravi. Improved approximation algorithms for requirement cut. *Operations Research Letters*, 38(4):322–325, 2010.

[GNRS04] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Cuts, trees and ℓ_1-embeddings of graphs. *Combinatorica*, 24(2):233–269, 2004.

[Gup01] A. Gupta. Steiner points in tree metrics don't (really) help. In *Proc. 12th SODA*, pages 220–227, 2001.

[KPR93] P. Klein, S. A. Plotkin, and S. B. Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proc. 25th STOC*, pages 682–690, 1993.

[LM10] T. Leighton and A. Moitra. Extensions and limits to vertex sparsification. In *Proc. 42th STOC*, pages 47–56, 2010.

[LN05] J. R. Lee and A. Naor. Extending Lipschitz functions via random metric partitions. *Invent. Math.*, 160(1):59–95, 2005.

[LS09] J. R. Lee and A. Sidiropoulos. On the geometry of graphs with a forbidden minor. In *Proc. 41st STOC*, pages 245–254, 2009.

[Moi09] A. Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *Proc. 50th FOCS*, pages 3–12, 2009.

[NL09] S. Nowozin and C. H. Lampert. Global connectivity potentials for random field models. In *Proc. 22nd CVPR*, pages 818–825, 2009.

[Räc08] H. Räcke. Optimal hierarchical decompositions for congestion minimization in net works. In *Proc. 40th STOC*, pages 255–264, 2008.

[VKR08] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *Proc. 21st CVPR*, 2008.