

# Cutting corners cheaply, or how to remove Steiner points

Lior Kamma\*

Robert Krauthgamer\*

Huy L. Nguyễn<sup>†</sup>

April 4, 2013

## Abstract

Our main result is that the Steiner Point Removal (SPR) problem can always be solved with polylogarithmic distortion, which resolves in the affirmative a question posed by Chan, Xia, Konjevod, and Richa (2006). Specifically, we prove that for every edge-weighted graph  $G = (V, E, w)$  and a subset of terminals  $T \subseteq V$ , there is a graph  $G' = (T, E', w')$  that is isomorphic to a minor of  $G$ , such that for every two terminals  $u, v \in T$ , the shortest-path distances between them in  $G$  and in  $G'$  satisfy  $d_{G,w}(u, v) \leq d_{G',w'}(u, v) \leq O(\log^6 |T|) \cdot d_{G,w}(u, v)$ . Our existence proof actually gives a randomized polynomial-time algorithm.

Our proof features a new variant of metric decomposition. It is well-known that every finite metric space  $(X, d)$  admits a  $\beta$ -separating decomposition for  $\beta = O(\log |X|)$ , which roughly means for every desired diameter bound  $\Delta > 0$  there is a randomized partitioning of  $X$ , which satisfies the following separation requirement: for every  $x, y \in X$ , the probability they lie in different clusters of the partition is at most  $\beta d(x, y)/\Delta$ . We introduce an additional requirement, which is the following tail bound: for every shortest-path  $P$  of length  $d(P) \leq \Delta/\beta$ , the number of clusters of the partition that meet the path  $P$ , denoted  $Z_P$ , satisfies  $\Pr[Z_P > t] \leq 2e^{-\Omega(t)}$  for all  $t > 0$ .

---

\*Weizmann Institute of Science, {lior.kamma, robert.krauthgamer}@weizmann.ac.il. This work was supported in part by a US-Israel BSF grant #2010418, and by the Citi Foundation. Part of this work was done while visiting Microsoft Research New England.

<sup>†</sup>Princeton University, hlnguyen@princeton.edu. This work was supported in part by NSF CCF 0832797, and a Gordon Wu Fellowship. Part of this work was done while interning at Microsoft Research New England.

# 1 Introduction

*Graph compression* describes the transformation of a given graph  $G$  into a small graph  $G'$  that preserves certain features (quantities) of  $G$ , such as distances or cut values. Notable examples for this genre include graph spanners, distance oracles, cut sparsifiers, and spectral sparsifiers, see e.g. [PS89, TZ05, BK96, BSS09] and references therein. The algorithmic utility of such graph transformations is clear – once the “compressed” graph  $G'$  is computed (as a preprocessing step), further processing can be performed on  $G'$  instead of on  $G$ , using less resources like runtime and memory, or achieving better accuracy (when the solution is approximate). See more in Section 1.3.

Within this context, we study *vertex-sparsification*, where  $G$  has a designated subset of vertices  $T$ , and the goal is reduce the number of vertices in the graph while maintaining certain properties of  $T$ . A prime example for this genre is vertex-sparsifiers that preserve terminal versions of (multicommodity) cut and flow problems, a successful direction that was initiated by Moitra [Moi09] and extended in several followups [LM10, CLLM10, MM10, EGK<sup>+</sup>10, Chu12]. Our focus here is different, on preserving distances, a direction that was seeded by Gupta [Gup01] more than a decade ago.

**Steiner Point Removal (SPR).** Let  $G = (V, E, w)$  be an edge-weighted graph<sup>1</sup> and let  $T = \{t_1, \dots, t_k\} \subseteq V$  be a designated set of  $k$  terminals. Here and throughout,  $d_{G,w}(\cdot, \cdot)$  denotes the shortest-path metric induced by  $w$  on the vertices of  $G$ . The *Steiner Point Removal* problem asks to construct on the terminals a new graph  $G' = (T, E', w')$  such that (i) distances between the terminals are *distorted* at most by factor  $\alpha \geq 1$ , formally

$$\forall u, v \in T, \quad d_{G,w}(u, v) \leq d_{G',w'}(u, v) \leq \alpha \cdot d_{G,w}(u, v);$$

and (ii) the graph  $G'$  is (isomorphic to) a minor of  $G$ . This formulation of the SPR problem was proposed by Chan, Xia, Konjevod, and Richa [CXKR06, Section 5] who posed the problem of bounding the distortion  $\alpha$  (existentially and/or using an efficient algorithm). Our main result is to resolve their open question.

Requirement (ii) above expresses structural similarity between  $G$  and  $G'$ ; for instance, if  $G$  is planar then so is  $G'$ . The SPR formulation above actually came about as a generalization to a result of Gupta [Gup01], which achieves  $\alpha = 8$  for the case where  $G$  is a tree,<sup>2</sup> and this factor of 8 was proved by [CXKR06] to be tight. The upper bound for trees was later extended by Basu and Gupta [BG08], who achieve distortion  $\alpha = O(1)$  for the larger class of outerplanar graphs.

**How to construct minors.** We now describe a general methodology that is natural for the SPR problem. The first step constructs a minor  $G'$  with vertex set  $T$ , but without any edge weights, and is prescribed by Definition 1.1. The second step determines edge weights  $w'$  such that  $d_{G',w'}$  dominates  $d_{G,w}$  on the terminals  $T$ , and is given in Definition 1.2. These steps are illustrated in Figure 1. Our definitions are actually more general (anticipating the technical sections), and consider  $G'$  whose vertex set is sandwiched between  $T$  and  $V$ . Define a *partial partition* of  $V$  to be a collection  $V_1, \dots, V_k$  of pairwise disjoint subsets of  $V$ .

---

<sup>1</sup>Throughout, all graphs are undirected and all edge weights are positive.

<sup>2</sup>In fact, Gupta [Gup01] only argued that  $G'$  is a tree. Chan *et al.* [CXKR06] observed later that this same  $G'$  is actually a minor of  $G$ .

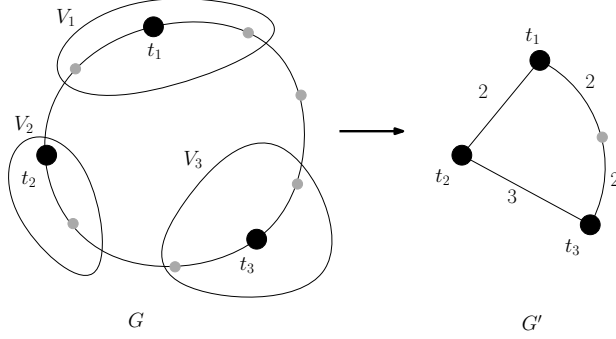


Figure 1:  $G$  is a 9-cycle with 3 terminals and unit edge weights. Its terminal-centered minor and the standard restriction edge weights are shown on the right.

**Definition 1.1** (Terminal-Centered Minor). *Let  $G = (V, E)$  be a graph with  $k$  terminals  $T = \{t_1, \dots, t_k\}$ , and let  $V_1, \dots, V_k$  be a partial partition of  $V$ , such that each induced subgraph  $G[V_j]$  is connected and contains  $t_j$ . The graph  $G' = (V', E')$  obtained by contracting each  $G[V_j]$  into a single vertex that is identified with  $t_j$ , is called the terminal-centered minor of  $G$  induced by  $V_1, \dots, V_k$ .*

By identifying the “contracted super-node”  $V_j$  with  $t_j$ , we may think of the vertex-set  $V'$  as containing  $T$  and (possibly) some vertices from  $V \setminus T$ . A terminal-centered minor  $G'$  of  $G$  can also be described by a mapping  $f : V \rightarrow T \cup \{\perp\}$ , such that  $f|_T \equiv \text{id}$  and  $f^{-1}(\{t_j\})$  is connected in  $G$  for all  $j \in [k]$ . Indeed, simply let  $V_j = f^{-1}(\{t_j\})$  for all  $j \in [k]$ , and thus  $V \setminus (\cup_j V_j) = f^{-1}(\{\perp\})$ .

**Definition 1.2** (Standard Restriction). *Let  $G = (V, E, w)$  be an edge-weighted graph with terminals set  $T$ , and let  $G' = (V', E')$  be a terminal-centered minor of  $G$ . The standard restriction of  $w$  to  $G'$  is the edge weight  $w'$  given by the respective distances in  $G$ , formally*

$$\forall (x, y) \in E', \quad w'_{xy} := d_{G,w}(x, y).$$

This edge weight  $w'$  is optimal in the sense that  $d_{G',w'}$  dominates  $d_{G,w}$  (where it is defined, i.e., on  $V'$ ), and the weight of each edge  $(x, y) \in E'$  is minimal under this domination condition.

## 1.1 Main Result

Our main result below gives an efficient algorithm that achieves  $\text{polylog}(k)$  distortion for the SPR problem. Its proof spans Sections 3 and 4, though the former one contains the heart of the matter.

**Theorem 1.3.** *Let  $G = (V, E, w)$  be an edge-weighted graph with  $k$  terminals  $T \subseteq V$ . Then there exists a terminal-centered minor  $G' = (T, E', w')$  of  $G$ , such that  $G'$  has distortion  $O(\log^6 k)$ , i.e.,*

$$\forall u, v \in T, \quad 1 \leq \frac{d_{G',w'}(u, v)}{d_{G,w}(u, v)} \leq O(\log^6 k).$$

Moreover,  $w'$  is the standard restriction of  $w$ , and  $G'$  is computable in randomized polynomial time.

This theorem answers a question of [CXKR06].<sup>3</sup> The only distortion lower bound known for general graphs is a factor of 8 (which actually holds for trees) [CXKR06], and thus it remains a challenging open question whether  $O(1)$  distortion can be achieved in general graphs.

Our proof of Theorem 1.3 begins similarly to [EGK<sup>+</sup>10], by iterating over the “distance scales”  $2^i$ , going from the smallest distance  $d_{G,w}(u, v)$  among all terminals  $u, v \in T$ , towards the largest such distance. Each iteration  $i$  first employs a “stochastic decomposition”, which is basically a randomized procedure that partitions  $V$  into so-called “clusters” (disjoint subsets) whose diameter is at most  $2^i$ . Then, some clusters are contracted to a nearby terminal, which must be “adjacent” to the cluster; this way, the current graph is a minor of the previous iteration’s graph, and thus also of the initial  $G$ . Let us now skip some technical details about the connectivity of the clusters, which is important for the contractions to be legit, and just say that at a high level, after iteration  $i$  is executed, we roughly expect “areas” of radius proportional to  $2^i$  around the terminals to be contracted. As  $i$  increases, these areas get larger until eventually all the vertices are contracted into terminals, at which point the weights are set to be standard restriction.

The main challenge is to control the distortion, and this is where we crucially deviate from [EGK<sup>+</sup>10] (and differ from all previous work). In their randomized construction of a minor  $G'$ , for every two terminals  $u, v \in T$  it is shown that  $G'$  contains a  $u - v$  path of *expected length* at most  $O(\log k)d_G(u, v)$ . Consequently, they design a *distribution*  $D$  over minors  $G'$ , such that the stretch  $d_{G'}(u, v)/d_G(u, v)$  between any  $u, v \in T$  has expectation at most  $O(\log k)$ .<sup>4</sup> In contrast, in our randomized construction of  $G'$ , the stretch between  $u, v \in T$  is polylogarithmic *with high probability*, say at least  $1 - 1/k^3$ . Applying a simple union bound over the  $\binom{k}{2}$  terminal pairs, we can then obtain a single graph  $G'$  achieving a polylogarithmic distortion. Technically, these bounds follow by fixing in  $G$  a shortest-path  $P$  between two terminals  $u, v \in T$ , and then tracking the execution of the randomized algorithm to analyze how the  $P$  evolves into a  $u - v$  path  $P'$  in  $G'$ . In [EGK<sup>+</sup>10], the length of  $P'$  is analyzed in expectation, which by linearity of expectation, follows from analyzing a single edge. In contrast, we provide for it a high-probability bound, which inevitably must consider (anti)correlations along the path.

The next section features a new tool that we developed in our quest for high-probability bounds, and which may be of independent interest. For sake of clarity, we provide below a vanilla version that excludes technical complications such as terminals, strong diameter, and consistency between scales. The proof of Theorem 1.3 actually does require these complications, and thus cannot use the generic form described below.

## 1.2 A Key Technique: Metric Decomposition with Concentration

**Metric decomposition.** Let  $(X, d)$  be a metric space, and let  $\Pi$  be a partition of  $X$ . Every  $S \in \Pi$  is called a *cluster*, and for every  $x \in X$ , we use  $\Pi(x)$  to denote the unique cluster  $S \in \Pi$  such that  $x \in S$ . In general, a stochastic decomposition of the metric  $(X, d)$  is a distribution  $\mu$  over partitions of  $X$ , although we usually impose additional requirements. The following definition is perhaps the most basic version, often called a separating decomposition or a Lipschitz decomposition.

**Definition 1.4.** *A metric space  $(X, d)$  is called  $\beta$ -decomposable if for every  $\Delta > 0$  there is a probability distribution  $\mu$  over partitions of  $X$ , satisfying the following requirements:*

<sup>3</sup>We remark that distortion  $k$  can be achieved relatively easily, by considering a mapping  $f$  that sends every vertex into its nearest terminal, and taking the minor corresponding to  $f$  with standard-restriction edge weights.

<sup>4</sup>But it is possible that no  $G' \in \text{supp}(D)$  achieves a low stretch simultaneously for all  $u, v \in T$ .

- (a). *Diameter bound:* for all  $\Pi \in \text{supp}(\mu)$  and all  $S \in \Pi$ ,  $\text{diam}(S) \leq \Delta$ .
- (b). *Separation probability:* for all  $x, y \in X$ ,  $\Pr_{\Pi \sim \mu} [\Pi(x) \neq \Pi(y)] \leq \frac{\beta d(x, y)}{\Delta}$ .

Bartal [Bar96] proved that every  $n$ -point metric is  $O(\log n)$ -decomposable, and that this bound is tight. We remark that by now there is a rich literature on metric decompositions, and different variants of this notion may involve terminals, or (in a graphical context) connectivity requirements inside each cluster, see e.g. [Bar96, CKR01, FRT04, Bar04, LN05, GNR10, EGK<sup>+</sup>10, MN07, AGMW10, KR11].

**Degree of separation.** Let  $P = (x_0, x_1, \dots, x_\ell)$  be a *shortest path* between  $x_0, x_\ell \in X$ , i.e., a sequence of points in  $X$  such that  $\sum_{i \in [\ell]} d(x_{i-1}, x_i) = d(x_0, x_\ell)$ . We denote its length by  $d(P) := d(x_0, x_\ell)$ , and say that  $P$  *meets* a cluster  $S \subseteq X$  if  $S \cap P \neq \emptyset$ . Given a partition  $\Pi$  of  $X$ , define the *degree of separation*  $Z_P(\Pi)$  as the number of different clusters in the partition  $\Pi$  that meet  $P$ . Formally

$$Z_P(\Pi) := \sum_{S \in \Pi} \mathbb{1}_{\{P \text{ meets } S\}}. \quad (1)$$

Throughout, we omit the partition  $\Pi$  when it is clear from the context. When we consider a random partition  $\Pi \sim \mu$ , the corresponding  $Z_P = Z_P(\Pi)$  is actually a random variable. If this distribution  $\mu$  satisfies requirement (b) of Definition 1.4, then

$$\mathbb{E}_{\Pi \sim \mu} [Z_P] \leq 1 + \sum_{i \in [\ell]} \Pr_{\Pi \sim \mu} [\Pi(x_{i-1}) \neq \Pi(x_i)] \leq 1 + \sum_{i \in [\ell]} \frac{\beta d(x_{i-1}, x_i)}{\Delta} = 1 + \frac{\beta d(P)}{\Delta}. \quad (2)$$

But what about the concentration of  $Z_P$ ? More precisely, can every finite metric be decomposed, such that every shortest path  $P$  admits a tail bound on its degree of separation  $Z_P$ ?

**A tail bound.** We answer this last question in the affirmative using following theorem. We prove it, or actually a stronger version that does involve terminals, in Section 2.

**Theorem 1.5.** *For every  $n$ -point metric space  $(X, d)$  and every  $\Delta > 0$  there is a probability distribution  $\mu$  over partitions of  $X$  that satisfies, for  $\beta = O(\log n)$ , requirements (a)-(b) of Definition 1.4, and furthermore*

- (c). *Degree of separation:* For every shortest path  $P$  of length  $d(P) \leq \frac{\Delta}{\beta}$ ,

$$\forall t \geq 1, \quad \Pr_{\Pi \sim \mu} [Z_P > t] \leq 2e^{-\Omega(t)}. \quad (3)$$

The tail bound (3) can be compared to a naive estimate that holds for every  $\beta$ -decomposition  $\mu$ : using (2) we have  $\mathbb{E}[Z_P] \leq 2$ , and then by Markov's inequality  $\Pr[Z_P \geq t] \leq 2/t$ .

### 1.3 Related Work

**Applications.** Vertex-sparsification, and the “graph compression” approach in general, is obviously beneficial when  $G'$  can be computed from  $G$  very efficiently, say in linear time, and then  $G'$  may be computed on the fly rather than in advance. But compression may be valuable also in scenarios that require the storage of many graphs, like archiving and backups, or rely on low-throughput communication, like distributed or remote processing. For instance, the succinct nature of  $G'$  may be indispensable for computations performed frequently, say on a smartphone, with preprocessing done in advance on a powerful machine.

We do not have new theoretical applications that leverage our SPR result, although we anticipate these will be found later. Either way, we believe this line of work will prove technically productive, and may influence, e.g., work on metric embeddings and on approximate min-cut/max-flow theorems.

**Probabilistic SPR.** Here, the objective is not to find a single graph  $G' = (T, E', w')$ , but rather a distribution  $D$  over graphs  $G' = (T, E', w')$ , such that every graph  $G' \in \text{supp}(D)$  is isomorphic to a minor of  $G$  and its distances  $d_{G',w'}$  dominate  $d_{G,w}$  (on  $T \times T$ ), and such that the distortion inequalities hold in expectation, that is,

$$\forall u, v \in T, \quad \mathbb{E}_{G' \sim D} [d_{G',w'}(u, v)] \leq \alpha \cdot d_{G,w}(u, v).$$

This problem, first posed in [CXKR06], was answered in [EGK<sup>+</sup>10] with  $\alpha = O(\log |T|)$ .

**Distance Preserving Minors.** This problem differs from SPR in  $G'$  may contain a few non-terminals, but all terminal distances should be preserved exactly. Formally, the objective is to find a small graph  $G' = (V', E', w')$  such that (i)  $G'$  is isomorphic to a minor of  $G$ ; (ii)  $T \subseteq V' \subseteq V$ ; and (iii) for every  $u, v \in T$ ,  $d_{G',w'}(u, v) = d_{G,w}(u, v)$ . This problem was originally defined by Krauthgamer and Zondiner [KZ12], who showed an upper bound  $|V'| \leq O(|T|^4)$  for general graphs, and a lower bound of  $\Omega(|T|^2)$  that holds even for planar graphs.

## 2 Metric Decomposition with Concentration

We now prove a slightly stronger result than that of Theorem 1.5, stated as Theorem 2.2. Let  $(X, d)$  be a metric space, and let  $\{t_1, \dots, t_k\} \subseteq X$  be a designated set of terminals. Recall that a partial partition  $\Pi$  of  $X$  is a collection of pairwise disjoint subsets of  $X$ . For a shortest path  $P$  in  $X$ , define  $Z_P = Z_P(\Pi)$  using Eqn. (1), which is similar to before, except that now  $\Pi$  is a partial partition. We first extend Definition 1.4.

**Definition 2.1.** *We say that  $X$  is  $\beta$ -terminal-decomposable with concentration if for every  $\Delta > 0$  there is a probability distribution  $\mu$  over partial partitions of  $X$ , satisfying the following properties.*

- **Diameter Bound:** For all  $\Pi \in \text{supp}(\mu)$  and all  $S \in \Pi$ ,  $\text{diam}(S) \leq \Delta$ .
- **Separation Probability:** For every  $x, y \in X$ ,

$$\Pr_{\Pi \sim \mu} [\exists S \in \Pi \text{ such that } |S \cap \{x, y\}| = 1] \leq \frac{\beta d(x, y)}{\Delta}.$$

- Terminal Cover: For every  $\Pi \in \text{supp}(\mu)$ ,  $T \subseteq \bigcup_{S \in \Pi} S$ .
- Degree of Separation: For every shortest path  $P$  and for every  $t \geq 1$ ,

$$\Pr_{\Pi \sim \mu} \left[ Z_P > \frac{t\beta}{\Delta} \max\{d(P), \Delta/\beta\} \right] \leq O \left( \min \left\{ k\beta, \left\lceil \frac{d(P)\beta}{\Delta} \right\rceil \right\} \right) e^{-\Omega(t)}$$

**Theorem 2.2.** *Every finite metric space with  $k$  terminals is  $(4 \log k)$ -terminal-decomposable with concentration.*

Define the *truncated exponential* with parameters  $\lambda, \Delta > 0$ , denoted  $\text{Texp}(\lambda, \Delta)$ , to be distribution given by the probability density function  $g_{\lambda, \Delta}(x) = \frac{1}{\lambda(1-e^{-\Delta/\lambda})} e^{-x/\lambda}$  for all  $x \in [0, \Delta]$ .

We are now ready to prove Theorem 2.2. For simplicity of notation, we prove the result with cluster diameter at most  $2\Delta$  instead of  $\Delta$ . Fix a desired diameter bound  $\Delta > 0$ , and set for the rest of the proof  $\lambda := \frac{\Delta}{\log k}$  and  $g := g_{\lambda, \Delta}$ . For  $x \in X$  and  $r > 0$ , we use the standard notation of a closed ball  $B(x, r) := \{y \in X : d(x, y) \leq r\}$ . We define the distribution  $\mu$  via the following procedure that samples a partial partition  $\Pi$  of  $X$ .

- 1: **for**  $j = 1, 2, \dots, k$  **do**
- 2:   choose independently at random  $R_j \sim \text{Texp}(\lambda, \Delta)$ , and let  $B_j = B(t_j, R_j)$ .
- 3:   set  $S_j = B_j \setminus \bigcup_{m=1}^{j-1} B_m$ .
- 4: **return**  $\Pi = \{S_1, \dots, S_k\} \setminus \{\emptyset\}$ .

The diameter bound and terminal partition properties hold by construction. The proof of the separation event property is identical to the one in [Bar96, Section 3]. The following two lemmas prove the degree of separation property, which will conclude the proof of Theorem 2.2. Fix a shortest path  $P$  in  $X$ , and let us assume that  $t/2$  is a positive integer; a general  $t \geq 1$  can be reduced to this case up to a loss in the unspecified constant.

**Lemma 2.3.** *If  $d(P) < \lambda$ , then  $\Pr[Z_P > t] \leq 2e^{-\Omega(t)}$ .*

*Proof.* Split the  $k$  terminals into  $J_{\text{far}} := \{j \in [k] : d(t_j, P) > \Delta - 2\lambda\}$  and  $J_{\text{near}} := [k] \setminus J_{\text{far}}$ . Define random variables  $Z_{\text{far}} := \#\{j \in J_{\text{far}} : B_j \cap P \neq \emptyset\}$  and  $Z_{\text{near}} := \#\{j \in J_{\text{near}} : S_j \cap P \neq \emptyset\}$ . Then  $Z_P \leq Z_{\text{far}} + Z_{\text{near}}$  and

$$\Pr[Z_P > t] \leq \Pr[Z_{\text{far}} + Z_{\text{near}} > t] \leq \Pr[Z_{\text{far}} > t/2] + \Pr[Z_{\text{near}} > t/2].$$

For every  $j \in J_{\text{far}}$ ,

$$\Pr[B_j \cap P \neq \emptyset] \leq \Pr[R_j \geq \Delta - 2\lambda] = \int_{\Delta-2\lambda}^{\Delta} g(x) dx = \frac{k}{k-1} (e^{-\frac{\Delta-2\lambda}{\lambda}} - e^{-\frac{\Delta}{\lambda}}) \leq \frac{8}{k},$$

and therefore  $\mathbb{E}[Z_{\text{far}}] \leq 8$ . Since  $Z_{\text{far}}$  is the sum of independent indicators, by a Chernoff bound  $\Pr[Z_{\text{far}} > t/2] \leq 2^{-t/2}$  for all  $t \geq 32e$ . For smaller  $t$ , observe that  $\Pr[Z_{\text{far}} = 0] \geq (1 - 8/k)^k \geq \Omega(1)$ , and thus for every  $t \geq 1$  we have  $\Pr[Z_{\text{far}} > t/2] \leq e^{-\Omega(t)}$ .

Next, consider which balls among  $\{B_j : j \in J_{\text{near}}\}$  have non-empty intersection with  $P$ , and let  $j_1 < j_2 < \dots$  denote their indices. Formally, we place a conditioning henceforth on a some event  $\mathcal{E}$

that determines whether  $R_j \geq d(t_j, P)$  occurs or not for each  $j \in J_{\text{near}}$ . For  $a = 1, 2, \dots$ , let  $Y_a$  be an indicator for the event that ball  $B_{j_a}$  does *not* contain  $P$ . Then

$$\Pr[Y_a = 1 \mid \mathcal{E}] = \Pr \left[ P \not\subseteq B_{j_a} \mid P \cap B_{j_a} \neq \emptyset \right] \leq \Pr \left[ R_j < d(t_{j_a}, P) + \lambda \mid R_j > d(t_{j_a}, P) \right] \leq \frac{1 - e^{-1}}{1 - e^{-2}} \leq \frac{3}{4}.$$

Having conditioned on  $\mathcal{E}$ , the event  $\{Z_{\text{near}} > t/2\}$  implies that  $Y_a = 1$  for all  $a \in [t/2]$ , and since these random variables  $Y_a$  are independent,  $\Pr[Z_{\text{near}} > t/2 \mid \mathcal{E}] \leq (3/4)^{t/2} \leq e^{-Ct}$  for an appropriate constant  $C > 0$ . The last inequality holds for all such events  $\mathcal{E}$  (with the same constant  $C > 0$ ), and thus also without any such conditioning.

Altogether, we conclude that  $\Pr[Z_P > t] \leq 2e^{-\Omega(t)}$ .  $\square$

**Lemma 2.4.** *If  $d(P) \geq \lambda$ , then  $\Pr[Z_P > td(P)/\lambda] \leq O\left(\min\left\{k \log k, \left\lceil \frac{d(P)}{\lambda} \right\rceil\right\}\right) e^{-\Omega(t)}$ .*

*Proof.* Treating  $P$  as a continuous path, subdivide it into  $r := \lceil d(P)/\lambda \rceil$  segments, say segments of equal length that are (except for the last one) half open and half closed. The induced subpaths  $P_1, \dots, P_r$  of  $P$  are disjoint (as subsets of  $X$ ) and have length at most  $\lambda$  each, though some of subpaths may contain only one or even zero points of  $X$ . Writing  $Z_P = \sum_{i \in [r]} Z_{P_i}$ , we can apply a union bound and then Lemma 2.3 on each  $P_i$ , to obtain

$$\Pr[Z_P > td(P)/\lambda] \leq \Pr \left[ \exists i \in [r] \text{ such that } Z_{P_i} > t/2 \right] \leq O\left(\left\lceil \frac{d(P)}{\lambda} \right\rceil\right) \cdot e^{-\Omega(t)}.$$

Furthermore, for every  $j \in [k]$ , let  $\mathcal{A}_j := \{i \in [r] : P_i \cap B(t_j, \Delta) \neq \emptyset\}$ , and since  $P$  is a shortest path,  $|\mathcal{A}_j| \leq 4\Delta/\lambda = 4 \log k$ . Observe that  $Z_{P_i} = 0$  (with certainty) for all  $i \notin \cup_j \mathcal{A}_j$ , hence

$$\Pr[Z_P > td(P)/\lambda] \leq \Pr \left[ \exists i \in \cup_{j \in [k]} \mathcal{A}_j \text{ such that } Z_{P_i} > t/2 \right] \leq 4k \log k e^{-\Omega(t)}.$$

$\square$

### 3 Terminal-Centered Minors: Main Construction

This section proves Theorem 1.3 when  $\mathcal{D} := \frac{\max_{u,v \in T} d_G(u,v)}{\min_{u,v \in T} d_G(u,v)}$  satisfies the following assumption.

**Assumption 3.1.**  $\mathcal{D} \leq 2^{k^3}$ .

By scaling all edge weights, we may further assume that  $\min_{u,v \in T} d_G(u,v) = 1$ .

**Notation 1.** Let  $V_1, \dots, V_k \subseteq V$ . For  $S \subseteq [k]$ , denote  $V_S := \bigcup_{j \in S} V_j$ . In addition, denote  $V_\perp := V \setminus V_{[k]}$  and  $V_{\perp+j} := V_\perp \cup V_j$  for all  $j \in [k]$ .

We now present a randomized algorithm that, given a graph  $G = (V, E, w)$  and terminals  $T \subset V$ , constructs a terminal-centered minor  $G'$  as stated in Theorem 1.3. This algorithm maintains a partial partition  $\{V_1, V_2, \dots, V_k\}$  of  $V$ , starting with  $V_j = \{t_j\}$  for all  $j \in [k]$ . The sets grow monotonically during the execution of the algorithm. We may also think of the algorithm as if it maintains a mapping  $f : V \rightarrow T \cup \{\perp\}$ , starting with  $f|_T = \text{id}$  and gradually assigning a value in  $T$  to additional vertices, which correspond to the set  $V_{[k]}$ . Thus, we will also refer to the vertices in  $V_{[k]}$  as *assigned*, and to vertices in  $V_\perp$  as *unassigned*. The heart of the algorithm is



two nested loops (lines 4-9). During every iteration of the outer loop, the inner loop performs  $k$  iterations, one for every terminal  $t_j$ . Every inner-loop iteration picks a random radius (from an exponential distribution) and “grows”  $V_j$  to that radius (but without overlapping any other set). Every outer-loop iteration increases the expectation of the radius distribution. Eventually, all nodes are assigned, i.e.  $\{V_1, V_2, \dots, V_k\}$  is a partition of  $V$ .

**Definition 3.2.** Let  $U \subseteq V$ , by  $G[U]$  we denote the subgraph of  $G$  induced by  $U$ , with induced edge lengths (i.e.  $w|_{E(G[U])}$ ). For a subgraph  $H$  of  $G$  with induced edge lengths, a vertex  $v \in V(H)$  and  $r > 0$ , denote  $B_H(v, r) := \{u \in V(H) : d_H(u, v) \leq r\}$ , where  $d_H$  is the shortest path metric in  $H$  induced by  $w$ .

**Input:**  $G = (V, E, w)$ ,  $T = \{t_1, \dots, t_k\} \subseteq V$   
**Output:** A partition  $\{V_1, V_2, \dots, V_k\}$  of  $V$ .

- 1: set  $b \leftarrow 1 + 1/(35 \log k)$
- 2: for every  $j \in [k]$  set  $V_j \leftarrow \{t_j\}$ ,  $r_j = 0$ .
- 3: set  $i \leftarrow 0$ . //  $i$  is the iteration number of the outer loop.
- 4: **while**  $V_{[k]} \neq V$  **do**
- 5:      $i \leftarrow i + 1$ .
- 6:     **for all**  $j \in [k]$  **do**
- 7:         choose independently at random  $R_j^i \sim \exp(b^i)$ .
- 8:          $r_j \leftarrow r_j + R_j^i$ .
- 9:          $V_j \leftarrow V_j \cup B_{G[V_{\perp+j}]}(t_j, r_j)$ . // Actually, this is the same as  $V_j \leftarrow B_{G[V_{\perp+j}]}(t_j, r_j)$ .
- 10: **return**  $\{V_1, V_2, \dots, V_k\}$ .

**Algorithm 1:** Partitioning  $V$

**Claim 3.3.** The following properties hold throughout the execution of the algorithm.

1. For all  $j \in [k]$ ,  $V_j$  is connected in  $G$ , and  $t_j \in V_j$ .
2. For every  $j_1, j_2 \in [k]$ , if  $j_1 \neq j_2$ , then  $V_{j_1} \cap V_{j_2} = \emptyset$ .
3. For every outer loop iteration  $i$  and every  $j \in [k]$ , if  $V_j^i$  denotes the set  $V_j$  in the beginning of the  $i$ th iteration (of the outer loop), and  $V_j^i$  denotes the set  $V_j$  at the end of that iteration, then  $V_j^i \subseteq V_j^i$ .

In what follows, we analyze the stretch in distance between a fixed pair of terminals. We show that with probability at least  $1 - 1/k^3$ , the distance between these terminals in  $G'$  is at most  $O(\log^6 k)$  times their distance in  $G$ . By a union bound over all  $\binom{k}{2}$  pairs of terminals, we deduce Theorem 1.3. Let  $s, t \in T$ , and let  $P^*$  be a shortest  $st$ -path in  $G$ . Due to the triangle inequality, we may focus on pairs which satisfy  $V(P^*) \cap T = \{s, t\}$ , where  $V(P^*)$  is the node set of  $P^*$ . We denote  $\ell := w(P^*) = d_{G,w}(s, t)$ .

### 3.1 High Level Analysis

Following an execution of the algorithm, we maintain a (dynamic) path  $P$  between  $s$  and  $t$ . In a sense, in every step of the algorithm,  $P$  simulates an  $st$ -path in the terminal-centered minor induced by  $V_1, V_2, \dots, V_k$ . In the beginning of the execution, set  $P$  to be simply  $P^*$ . During the course

of the execution update  $P$  to satisfy two invariants: At every step of the algorithm, the weight of  $P$  is an upper bound on the distance between  $s$  and  $t$  in the terminal centered minor induced by  $V_1, \dots, V_k$  (in that step). In addition, if  $I$  is a subpath of  $P$ , whose inner vertices are all unassigned, then  $I$  is a subpath of  $P^*$ . Throughout the analysis, we think of  $P$  as directed from  $s$  to  $t$ , thus inducing a linear ordering of the vertices in  $P$ .

**Definition 3.4.** *A subpath of  $P$  will be called active if it is a maximal subpath whose inner vertices are unassigned.*

We now describe how  $P$  is updated during the execution of the algorithm. Consider line 9 of the algorithm for the  $i$ th iteration of the outer loop, and some  $j \in [k]$ . We say that the ball  $B = B_{G[V_{\perp+j}]}(t_j, r_j)$  punctures an active subpath  $A$  of  $P$ , if there is an inner node of  $A$  that belongs to the ball. If  $B$  does not puncture any active subpath of  $P$ , we do not change  $P$ . Otherwise, denote by  $u, v$  the first and last unassigned nodes (possibly not in the same active subpath) in  $V(P) \cap B$  respectively. Then we do the following:

1. We replace the entire subpath of  $P$  between  $u$  and  $v$  with a concatenation of a shortest  $ut_j$ -path and a shortest  $t_jv$ -path that lie in  $B$ ; this is possible, since  $G[B]$  is connected, and  $u, t_j, v \in B$ . This addition to  $P$  will be called a *detour* from  $u$  to  $v$  through  $t_j$ . The process is illustrated in figures 2(a)-2(c). Beginning with  $P^*$ , the figure describes the update after the first four balls. Note that the detour might not be a simple path. It is also worth noting that here  $u$  and  $v$  may belong to different active subpaths of  $P$ . For example, in figure 2(c), the new ball punctures two active subpaths, and therefore in figure 2(d), the detour goes from a node in one active subpath to a node in another active subpath. Note that in this case, we remove from  $P$  portions which are not active.
2. If, for some iteration  $i' < i$  of the outer loop, and for some  $u', v' \in V(P)$ , we added a detour from  $u'$  to  $v'$  through  $t_j$  in iteration  $i'$ , we keep only one detour through  $t_j$ , from the first node between  $u, u'$  and to the last between  $v, v'$ . For example, in figure 2(d), the ball centered in  $t_3$  punctures an active subpath. Only one detour is kept in figure 2(e).

The total weight of all detours during the execution will be called the *additional weight* to  $P$  (ignoring portions of  $P$  that are deleted from  $P$ ). Denote the set of active subpaths of  $P$  in the beginning of the  $i$ th iteration of the outer loop by  $\mathcal{A}_i$ .

Let  $V_1^{fin}, \dots, V_k^{fin}$  be the partition returned by the algorithm, let  $G'$  the terminal-centered minor induced by that partition, and let  $w'$  be the standard restriction of  $w$  to  $G'$ . Denote by  $P^{fin}$  the path obtained at the end of the execution.

**Claim 3.5.** *At every step of the algorithm the following holds:*

1. *The weight of  $P$  is an upper bound on the distance between  $s$  and  $t$  in the terminal centered minor induced by  $V_1, \dots, V_k$ . Moreover, once  $\mathcal{A}_i = \emptyset$  (namely,  $P$  has no active subpaths), the additional weight of  $P$  is an upper bound on the distance between  $s$  and  $t$  in the terminal centered minor induced by  $V_1^{fin}, \dots, V_k^{fin}$  (actually, from this point on,  $P = P^{fin}$ ).*
2. *If  $A$  is a subpath of  $P$ , whose inner points are all in  $V_{\perp}$ , then  $A$  is a subpath of  $P^*$ .*
3. *If  $A_1, A_2$  are two different active subpaths of  $P$ , they are internally disjoint.*
4.  *$|\mathcal{A}_i| \leq k$  for all  $i$ .*

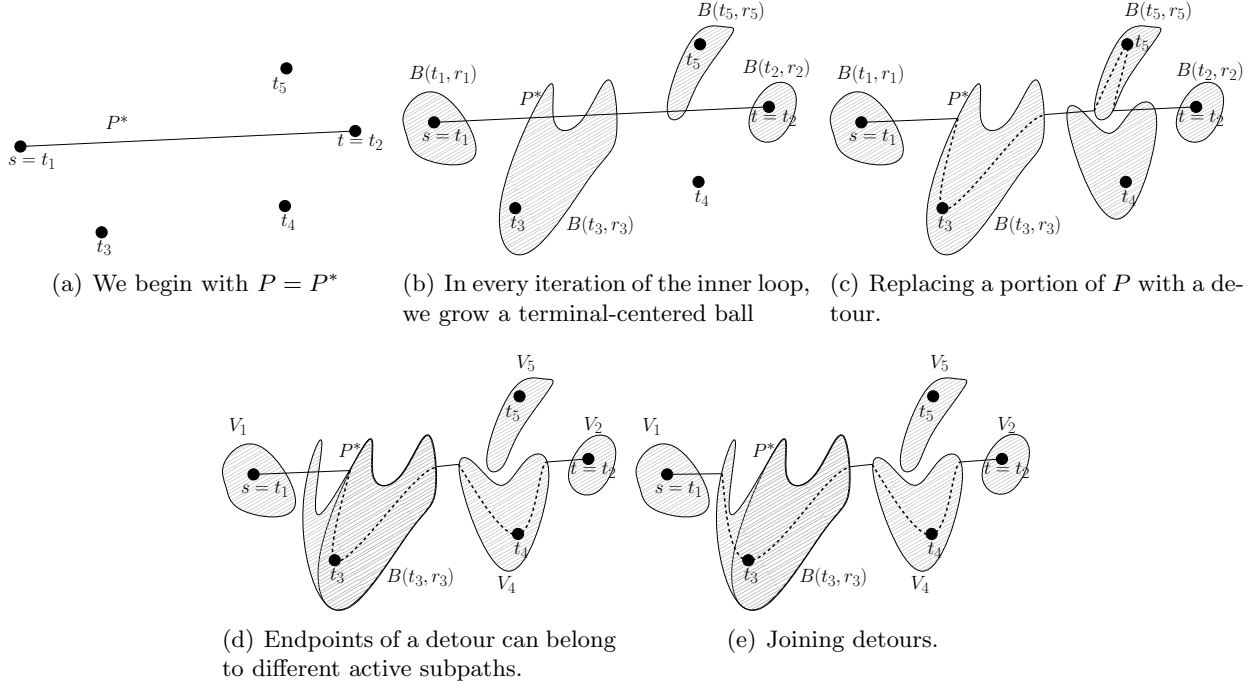


Figure 2: Updating  $P$

*Proof.* Follows easily by induction on  $i, j$ . □

**Corollary 3.6.**  $d_{G', w'}(s, t) \leq w(P^{fin})$ .

Let  $A \in \mathcal{A}_i$ . During the execution of the inner loop,  $A$  is either removed from  $P$  entirely, or some subpaths of  $A$  remain active (perhaps  $A$  remains active entirely). Therefore, for every  $A' \in \mathcal{A}_{i+1}$ , either  $A'$  is a non-trivial subpath of  $A$  (by non-trivial we mean  $|V(A')| \geq 3$ ), or  $A'$  and  $A$  are internally disjoint. Therefore there is a laminar structure on  $\bigcup_i \mathcal{A}_i$ .

We describe this structure using a tree  $\mathcal{T}$ , whose node set is  $\{\langle i, A \rangle \mid A \in \mathcal{A}_i\}$ . The root of  $\mathcal{T}$  is  $\langle 1, P^* \rangle$ , and for every  $i$  and every  $A \in \mathcal{A}_i$ , the children of  $\langle i, A \rangle$ , if any, are all pairs  $\langle i+1, A' \rangle$ , where  $A' \in \mathcal{A}_{i+1}$  is a subpath of  $A$ . Whenever we update  $P$  we log the weight of the detour by charging it to one of the nodes of  $\mathcal{T}$  as follows. Let  $u, v$  be unassigned nodes in  $P$ , and consider a detour from  $u$  to  $v$  in the  $i$ th iteration of the outer loop for some  $i$ . The weight of the detour is charged to the (unique) active subpath  $A \in \mathcal{A}_i$  such that  $u \in A$ . For every  $i$  and  $A \in \mathcal{A}_i$ , denote the total weight charged to  $\langle i, A \rangle$  by  $w_{i,A}$  (If the node is never charged, the weight of the node is set to 0). Therefore,

$$w(P^{fin}) \leq w(P^*) + \sum_{\langle i, A \rangle \in \mathcal{T}} w_{i,A} \tag{4}$$

Equation (4) together with Corollary 3.6 imply that if we show that with high probability, the total weight charged to the tree is at most  $O(\log^6 k) \cdot \ell$ , we can deduce Theorem 1.3. For the rest of this section, we therefore prove the following lemma.

**Lemma 3.7.** *With probability at least  $1 - O(k^{-3})$ , the total weight charged to the tree is at most  $O(\log^6 k)\ell$ .*

Let  $p = 1/100$ . An active subpath  $A \in \mathcal{A}_i$  will be called *short* if  $w(A) \leq pb^i$ . Otherwise,  $A$  will be called *long* (Note that  $P^* \in \mathcal{A}_1$  is long).

**Definition 3.8.** *Let  $i > 1$ . Let  $A \in \mathcal{A}_i$  be a short subpath. Denote by  $\mathcal{T}_{i,A}$  the subtree of  $\mathcal{T}$  rooted in  $\langle i, A \rangle$ . If, in addition, the parent of  $\langle i, A \rangle$  in  $\mathcal{T}$  consists of a long subpath,  $\mathcal{T}_{i,A}$  will be called a short subtree of  $\mathcal{T}$ .*

## 3.2 The Behavior of Short Subtrees

Clearly, every node of a short subtree of  $\mathcal{T}$  consists of a short active subpath. The height and weight of short subtrees will play an important role in the analysis of the height and weight of  $\mathcal{T}$ , and therefore we begin by analyzing short subtrees. Let  $i_0 > 1$ . Let  $A \in \mathcal{A}_{i_0}$  be a short subpath, and assume in addition that  $\mathcal{T}_{i_0,A}$  is a short subtree of  $\mathcal{T}$ .

### 3.2.1 The Effect of a Single Ball

Fix some  $i \geq i_0$ , and some  $j \in [k]$ . Let  $X$  denote the number of active subpaths of  $A$  in the beginning of the  $j$ th iteration of the inner loop (during the  $i$ th iteration of the outer loop). Let  $X'$  denote the number of active subpaths of  $A$  in the end of the  $j$ th iteration. Denote by  $B$  the ball considered in this iteration, namely  $B := B_{G[V_{\perp+j}]}(t_j, r_j)$ .

**Proposition 3.9.** *With certainty,  $X' \leq X + 1$ .*

*Proof.* Let  $A_1, A_2, \dots, A_X$  be the subpaths of  $A$  which are active in the beginning of the  $j$ th iteration ordered by their location on  $P$ . For  $\alpha \in [X]$  denote by  $u_\alpha, v_\alpha$  the first and last unassigned nodes in  $A_\alpha$ , respectively. If  $B$  does not puncture any of these subpaths, then  $X' \leq X < X + 1$  (Note that subpaths of  $A$  can still be removed if  $B$  punctures active subpaths of  $P$  not contained in  $A$ ). So assume  $B$  punctures  $A_\alpha$ . Assume first that  $A_\alpha$  is the only subpath of  $P$  which is active and is punctured by  $B$ . Then there are three options: If both  $u_\alpha, v_\alpha \in B$ , then  $A_\alpha$  is replaced and removed entirely from  $P$  when adding the detour, and  $X' \leq X - 1 < X + 1$ . If  $u_\alpha \in B$  and  $v_\alpha \notin B$ , let  $v'$  be the last node in  $V(A_\alpha) \cap B$  then the  $u_\alpha v'$  segment of  $A_\alpha$  is replaced, and the segment  $v'v_\alpha$  remains active. Therefore  $X' \leq X < X + 1$ . The argument is similar, if  $u_\alpha \notin B$  and  $v_\alpha \in B$ . Otherwise, some of the inner portion of  $A$  is replaced by a non active path, and both end segments of  $A_\alpha$  remain active, therefore  $X' = X + 1$ . Next, assume the ball punctures several active subpaths of  $A$ , and maybe more subpaths of  $P$ . Denote by  $I_\alpha, I_\beta$  the first and last subpaths of  $A$  punctured by  $B$ . Denote by  $u$  the first node in  $V(I_\alpha) \cap B$ , and  $v$  the last node in  $V(I_\beta) \cap B$ . When updating  $P$ , the entire subpath of  $P$  between  $u$  and  $v$  is removed. Thus  $X' \leq X - (\beta - \alpha + 1) \leq X < X + 1$ .  $\square$

For the next proposition, we need the following definition, which is a generalization of the notion of a ball puncturing an active subpath. Recall that a ball  $B$  is said to puncture an active subpath  $A'$  if there is an inner node in  $A'$  that belongs to the ball.

**Definition 3.10.** *Let  $A'$  be an active subpath of  $P$ . We say that a ball  $B$  reaches  $A'$  if  $V(A') \cap B \neq \emptyset$ .*

By the definition of the balls in the algorithm,  $B$  may reach  $A'$  and not puncture if and only if  $A'$  has one endpoint in  $V_j$ .

**Proposition 3.11.**

$$\Pr[X' \geq X \mid B \text{ reaches an active subpath of } A] \leq p.$$

*Proof.* Let  $A'$  be an active subpath of  $A$  closest to  $t_j$ . Let  $d = d_{G[V_{\perp+j}]}(t_j, A') = d_{G[V_{\perp+j}]}(t_j, A)$  be the distance in  $G[V_{\perp+j}]$  between  $t_j$  and a nearest node in  $A'$ . Assume that the ball  $B$  punctures an active subpath of  $A$ . Following the analysis of the previous proof, if  $X' \geq X$ , then  $B$  punctures exactly one active subpath of  $A$  and does not cover it, or  $B$  punctures exactly two active such subpaths and does not cover both of them. In either case,  $B$  punctures  $A'$  and does not cover it, implying that  $r_j + R_j^i \geq d$ , and  $r_j + R_j^i < d + pb^i$  (recall that  $A$  is short, and therefore, so is  $A'$ ). If  $B$  reaches an active subpath of  $A$ , and does not puncture it, we get the same conclusion, since this means  $r_j + R_j^i \geq r_j \geq d$ . By the memoryless property,

$$\begin{aligned} \Pr[X' \geq X \mid B \text{ reaches an active subpath of } A] &\leq \\ &\leq \Pr[R_j^i < d - r_j + pb^i \mid R_j^i \geq d - r_j] \leq 1 - e^{-p} \leq p \end{aligned}$$

□

**3.2.2 The Effect of a Sequence of Balls**

Consider the first  $N$  balls that reach some active subpath of  $A$ , starting from the beginning of iteration  $i_0$  of the outer loop, and perhaps during several iterations of that loop. For every  $a \in [N]$ , let  $Y_a$  be the indicator random variable for the event that the  $a$ th ball reaching an active subpath of  $A$  decreased the number of active subpaths. In these notations, Proposition 3.11 stated that

$$\forall a \in [N]. \Pr[Y_{a+1} = 1 \mid Y_1, \dots, Y_a] \geq 1 - p$$

Let  $Y = \sum_{a \in [N]} Y_a$  and let  $Z \sim \text{Bin}(N, 1 - p)$ . Simple induction on  $N$  implies the following claim.

**Claim 3.12.**  $\forall k. \Pr[Y > k] \geq \Pr[Z > k]$ .

**Lemma 3.13.** *With probability at least  $1 - 1/k^8$ , after  $70 \log k$  balls reach some active subpath of  $A$ ,  $A$  has no active subpath. In addition, the height of  $\mathcal{T}_{i_0, A}$  is at most  $35 \log k$ .*

*Proof.* Assume  $N = 70 \log k$ . Since whenever  $Y_a = 0$ , the number of active subpaths increases by at most 1, and whenever  $Y_a = 1$ , the number of active subpaths decreases by at least 1, if  $Y > N/2$ , then  $A$  has no active subpaths. Therefore by the Chernoff Bound,

$$\begin{aligned} \Pr[A \text{ has no active subpath after } N \text{ balls reach } A] &\geq \\ &\geq \Pr[Y > N/2] \geq \Pr[Z > N/2] \geq 1 - 1/k^8 \end{aligned}$$

Next, consider some  $i \geq i_0$ . If  $A$  has an active subpath at the end of the  $i$ th iteration of the outer loop, then at least two times during the  $i$ th iteration an active subpath of  $A$  is reached by a ball. After  $35 \log k$  iterations of the outer loop, if  $A$  has an active subpath, then  $N \geq 70 \log k$ . By the same arguments as the first part of the proof,

$$\Pr[\text{The height of } \mathcal{T}_{i_0, A} \text{ is at most } N/2] \geq \Pr[Y > N/2] \geq \Pr[Z > N/2] \geq 1 - 1/k^8$$

□

We denote by  $\mathbf{E}_1$  the event that for every  $i$ , and every  $A \in \mathcal{A}_i$ , if  $\mathcal{T}_{i,A}$  is a short subtree then after at most  $70 \log k$  balls reach an active subpath of  $A$ ,  $A$  has no more active subpaths and in addition, the height of  $\mathcal{T}_{i,A}$  is at most  $35 \log k$ .

**Lemma 3.14.**  $\Pr[\mathbf{E}_1] \geq 1 - 1/k^3$ .

*Proof.* Fix some  $i$ , and  $A \in \mathcal{A}_i$ . Assume that  $\mathcal{T}_{i,A}$  is a short subtree. By definition,  $\langle i, A \rangle$  has no short ancestor. For  $i' = \log_b(\ell/p) \leq \log_b(\mathcal{D}/p)$ ,  $P^*$  itself is short, since  $b^{i'} = \ell/p$ , and thus all tree nodes in level  $i'$  (and lower) are short. Therefore,  $i \leq i'$ . Since there are at most  $k$  nodes in every level of the tree, the number of short subtrees of  $\mathcal{T}$  is at most  $k \cdot \log_b(\mathcal{D}/p) = k \cdot (\log_b \mathcal{D} + \log_b 100) \leq k^4 \log k + O(k \log k) \leq O(k^4 \log k)$ . By the previous lemma, and a union bound over all short subtrees, the result follows.  $\square$

Since every node in level  $\log_b(\ell/p)$  of the tree belongs to some short subtree, we get the following corollary.

**Corollary 3.15.** *With probability at least  $1 - 2/k^3$ , the height of  $\mathcal{T}$  is at most  $\log_b(\ell/p) + 30 \log k \leq 10 \log_b \mathcal{D}$ .*

We denote by  $\mathbf{E}_2$  the event that for all  $i \leq 10 \log_b \mathcal{D}$  and  $j \in [k]$ , the radius of the  $j$ th ball of the  $i$ th iteration of the outer loop is at most  $280b^{i+1} \log^2 k$ .

**Lemma 3.16.**  $\Pr[\mathbf{E}_2] \geq 1 - 1/k^3$

*Proof.* Fix  $i \leq 10 \log_b \mathcal{D}$  and  $j \in [k]$ . Then  $\Pr[R_j^i > 8b^i \log k] \leq 1/k^8$ . By assumption 3.1,  $10 \log_b \mathcal{D} = O(\log \mathcal{D} \log k) = O(k^3 \log k)$ . Thus by a union bound over all values of  $i$  and  $j$  in question,

$$\Pr[\forall i, j. R_j^i \leq 8b^i \log k] \geq 1 - \frac{k \cdot 10 \log_b \mathcal{D}}{k^8} \geq 1 - \frac{1}{k^3}$$

It follows that, with probability at least  $1 - 1/k^3$ , the radius of the  $j$ th ball of the  $i$ th iteration of the outer loop is at most  $\sum_{i' \leq i} 8b^{i'} \log k \leq 280b^{i+1} \log^2 k$ .  $\square$

Summing everything up, we can now bound with high probability the weights of all short subtrees of  $\mathcal{T}$ .

**Claim 3.17.** *Conditioned on the events  $\mathbf{E}_1$  and  $\mathbf{E}_2$ , for every  $i_0 > 1$  and  $A \in \mathcal{A}_{i_0}$ . If  $\mathcal{T}_{i_0,A}$  is a short subtree of  $\mathcal{T}$ , then the total weight charged to nodes of  $\mathcal{T}_{i_0,A}$  is at most  $39200eb^{i_0+1} \log^3 k$  with certainty.*

*Proof.* Conditioned on  $\mathbf{E}_1$ , at most  $70 \log k$  detours are charged to nodes of every short subtree and for  $i = i_0 + 35 \log k$ , there are no more active subpaths of  $A$ . Conditioned on  $\mathbf{E}_2$ , the most expensive detour is of weight at most  $560b^{i_0+35 \log k+1} \log^2 k$ , we get that the total weight charged to nodes of the subtree is  $70 \log k \cdot 560b^{i_0+1} \cdot b^{35 \log k} \cdot \log^2 k \leq 39200eb^{i_0+1} \log^3 k$ .  $\square$

### 3.3 Bounding The Weight Of $\mathcal{T}$

We are now ready to bound the total weight charged to the tree. We sum the weights in each level of the tree from the root down. In each level we sum the weights charged to long active subpaths, and the weights of short subtrees rooted in that level. More formally, for every  $i \leq \log_b(\ell/p)$ ,

denote by  $l_i$  the total weight charged to nodes of the form  $\langle i, A \rangle$ , where  $A \in \mathcal{A}_i$  is a long active subpath. Denote by  $s_i$  the total weight charged to short subtrees rooted in the  $i$ th level of  $\mathcal{T}$ . For every  $i \geq \log_b(\ell/p)$ , and every  $A \in \mathcal{A}_i$ ,  $A$  is short and therefore  $\langle i, A \rangle$  belongs to some short subtree rooted in level at most  $\log_b(\ell/p)$ . Therefore,

$$\sum_{\langle i, A \rangle \in \mathcal{T}} w_{i, A} = \sum_{i=1}^{\log_b(\ell/p)} (l_i + s_i) \quad (5)$$

Since  $\ell \geq 1$ , then the root of  $\mathcal{T}$ , namely  $\langle 1, P^* \rangle$ , consists of a long path. Therefore,  $s_1 = 0$ . We can therefore rearrange Equation (5) to get the following.

$$\sum_{\langle i, A \rangle \in \mathcal{T}} w_{i, A} = \sum_{i=1}^{\log_b(\ell/p)} (l_i + s_{i+1}) \quad (6)$$

Let  $i \leq \log_b(\ell/p)$ . Let  $A \in \mathcal{A}_i$  be a long active subpath. That is  $w(A) \geq pb^i$ . Thinking of  $A$  as a continuous path, divide  $A$  into  $w(A)/(pb^i)$  segments of length  $pb^i$ . Some segments may contain no nodes. Let  $I$  be a segment of  $A$ , and assume  $I$  contains nodes (otherwise, no cost is charged to  $A$  on account of detours from  $I$ ). Following the same arguments of propositions 3.9 and 3.11, we get the following result.

**Lemma 3.18.** *With probability at least  $1 - 1/k^8$ , no more than  $70 \log k$  balls puncture  $I$ .*

Denote by  $\mathbf{E}_3$  the event that for every  $i \geq \log_b(\ell/k^2)$ , and for every long active subpath  $A \in \mathcal{A}_i$ , in the division of  $A$  to segments of length  $pb^i$ , every such subsegment is reached by at most  $70 \log k$  balls.

**Lemma 3.19.**  $\Pr[\mathbf{E}_3] \geq 1 - 1/k^3$ .

*Proof.* Since for every  $i \geq \log_b(\ell/p)$ , every  $A \in \mathcal{A}_i$  is short, the number of relevant iterations (of the outer loop) is at most  $\log_b(\ell/p) - \log_b(\ell/k^2) = \log_b(k^2/p) \leq O(\log^2 k)$ . For every  $i \geq \log_b(\ell/k^2)$  and every long path  $A \in \mathcal{A}_i$ , the number of segments of  $A$  is at most  $w(A)/(pb^i) \leq w(A)/(p\ell/k^2) \leq k^2/p$ . Therefore the number of relevant segments for all  $i \geq \log_b(\ell/k^2)$  and for all long  $A \in \mathcal{A}_i$  is at most  $O(k^2 \log^2 k)$ . Applying a union bound over all relevant segments the result follows.  $\square$

Since  $\Pr[\mathbf{E}_1] \geq 1 - 1/k^3$  and  $\Pr[\mathbf{E}_2] \geq 1 - 1/k^3$ , we get the following corollary.

**Corollary 3.20.**  $\Pr[\mathbf{E}_1 \wedge \mathbf{E}_2 \wedge \mathbf{E}_3] \geq 1 - O(k^{-3})$

It follows that it is enough for us to prove that conditioned on  $\mathbf{E}_1$ ,  $\mathbf{E}_2$  and  $\mathbf{E}_3$ , with probability 1 the total weight charged to the tree is at most  $O(\log^6 k)\ell$ .

**Lemma 3.21.** *Conditioned on  $\mathbf{E}_2$  and  $\mathbf{E}_3$ ,  $l_i \leq 560b^{i+1}k \log^2 k$ . In addition, if  $i \geq \log_b(\ell/k^2)$ ,  $l_i \leq O(\log^3 k) \cdot \ell$  with probability 1.*

*Proof.* To see the first bound, observe that by the update process of  $P$ , at most  $k$  detours are added to  $P$  during the  $i$ th iteration. Conditioned on  $\mathbf{E}_2$ , each one of them is of weight at most  $560b^{i+1} \log^2 k$ . To see the second bound, let  $A \in \mathcal{A}_i$  be a long active subpath. The additional

weight resulting from detours from vertices of  $A$  is at most the number of segments of  $A$  of length  $pb^i$ , times the additional weight to each segment. Therefore, the additional weight is at most

$$w_{i,A} \leq w(A)/pb^i \cdot 70 \log k \cdot 560b^{i+1} \log^2 k = 39200b/p \cdot \log^3 k \cdot w(A).$$

Since all paths in  $\mathcal{A}_i$  are internally disjoint subpaths of  $P^*$ , we get:

$$l_i = \sum_{\text{long } A \in \mathcal{A}_i} w_{i,A} \leq \sum_{\text{long } A \in \mathcal{A}_i} O(\log^3 k) \cdot w(A) \leq O(\log^3 k) \cdot \ell$$

□

**Lemma 3.22.** *Conditioned on events  $\mathbf{E}_1$ ,  $\mathbf{E}_2$  and  $\mathbf{E}_3$ ,  $s_{i+1} \leq 39200eb^{i+2}k \log^3 k$ . In addition, if  $i \geq \log_b(\ell/k^2)$ ,  $s_{i+1} \leq O(\log^4 k) \cdot \ell$ .*

*Proof.* Conditioned on  $\mathbf{E}_1$  and  $\mathbf{E}_2$ , we proved in Claim 3.17 that the total weight charged to a short subtree rooted in level  $i+1$  is at most  $39200eb^{i+2} \log^3 k$  with certainty. Since there are at most  $k$  such subtrees, the first bound follows. To get the second bound, note that by the definition of a short subtree, for every short subtree  $\mathcal{T}'$  rooted at level  $i+1$ , the parent of the root of  $\mathcal{T}'$  consists of a long active subpath  $A$  of level  $i$ . Conditioned on  $\mathbf{E}_3$ , every segment of  $A$  is intersected by at most  $70 \log k$  balls. Therefore,  $\langle i, A \rangle$  can have at most  $(w(A)/pb^i) \cdot 70 \log k$  children, and in particular, children consisting of short active subpaths. The cost of a short subtree rooted in the  $i+1$  level of  $\mathcal{T}$  is at most  $39200eb^{i+1} \log^3 k$ . Thus the total cost of all short subtrees rooted in children of  $A$  is bounded by

$$(w(A)/pb^i) \cdot 70 \log k \cdot 39200eb^{i+1} \log^3 k \leq O(\log^4 k) \cdot w(A)$$

Summing over all (internally disjoint) long subpaths of level  $i$ , the result follows. □

We now turn to the proof Lemma 3.7.

*Proof of Lemma 3.7.* Since  $\Pr[\mathbf{E}_1 \wedge \mathbf{E}_2 \wedge \mathbf{E}_3] \geq 1 - O(k^{-3})$ , it is enough to show that conditioned on  $\mathbf{E}_1$ ,  $\mathbf{E}_2$  and  $\mathbf{E}_3$ , the total weight charged to the tree is at most  $O(\log^6 k)\ell$  with certainty. Recall that  $\sum_{\langle i,A \rangle \in \mathcal{T}} w_{i,A} = \sum_{i=1}^{\log_b(\ell/p)} (l_i + s_{i+1})$ . Following Lemmas 3.21 and 3.22 we get that

$$\begin{aligned} \sum_{i=1}^{\log_b(\ell/k^2)} (l_i + s_{i+1}) &\leq \sum_{i=1}^{\log_b(\ell/k^2)} (560b^{i+1}k \log^2 k + 39200eb^{i+2}k \log^3 k) \\ &\leq O(k \log^3 k) \sum_{i=1}^{\log_b(\ell/k^2)} b^i = O(k \log^3 k) \frac{b}{b-1} \cdot \frac{\ell}{k^2} = o(1) \cdot \ell \end{aligned} \tag{7}$$

In addition,

$$\begin{aligned} \sum_{i=\log_b(\ell/k^2)+1}^{\log_b(\ell/p)} (l_i + s_{i+1}) &\leq \sum_{i=\log_b(\ell/k^2)+1}^{\log_b(\ell/p)} O(\log^3 k)\ell + O(\log^4 k)\ell \\ &\leq O(\log^4 k)\ell \cdot (\log_b(\ell/p) - \log_b(\ell/k^2)) \\ &\leq O(\log^4 k)\ell \cdot O(\log^2 k) = O(\log^6 k) \cdot \ell \end{aligned} \tag{8}$$

□



Before describing the generalized algorithm, used to discard Assumption 3.1, we wish to note the following fact, which follows directly from the analysis of Algorithm 1.

**Remark 1.** *By slightly changing the analysis of Algorithm 1 (without changing the actual algorithm) we can get that for every  $\tilde{k} \geq k$ , and every  $d \geq 3$ , the algorithm achieves a stretch factor of  $O(d^2 \log^6 \tilde{k})$  with probability  $1 - O(\tilde{k}^{-d})$ .*

## 4 Terminal-Centered Minors: Extension to General Case

In this section, we complete the proof of Theorem 1.3 by reducing it to the special case where Assumption 3.1 holds (which is proven in Section 3). Let us outline the recursive algorithm for the general case. The algorithm first rescales edge weights of the graph so that minimal terminal distance is 1. If  $\mathcal{D} < 2^{k^3}$  then we apply Algorithm 1 and we are done. Otherwise, we construct a set of at most  $k - 1$  well-separated and low diameter balls whose union contains all terminals. Then, for each of the balls, we apply Algorithm 1 on the graph induced by that ball. Each ball is then contracted into a “super-terminal”. We apply the algorithm recursively on the resulting graph  $\tilde{G}$  with the set of super-terminals as the terminal set. Going back from the recursion, we “stitch” together the output of Algorithm 1 on the balls in the original graph with the output of the recursive call on  $\tilde{G}$ , to construct a partition of  $V$  as required. The exact algorithm is described in Section 4.1. Before that, we need a few definitions.

Assume that the edge weights are already so that the minimum terminal distance is 1. Denote by  $D$  the set of all distances between terminals, rounded down to the nearest powers of 2. Note that  $|D| < k^2$ . Consider the case  $\mathcal{D} > 2^{k^3}$ . There must exist  $0 \leq m_0 \leq k^3 - k$  such that  $D \cap \{2^{m_0}, 2^{m_0+1}, \dots, 2^{m_0+k}\} = \emptyset$ . Define  $R := \{(x, y) \in T^2 : d_G(x, y) < 2^{m_0}\}$ .

**Claim 4.1.**  *$R$  is an equivalence relation.*

*Proof.* Reflexivity and symmetry of  $R$  follow directly from the definition of a metric. To see that  $R$  is transitive, let  $x, y, z \in T$ , and assume  $(x, y), (y, z) \in R$ . Therefore  $d_G(x, y) < 2^{m_0}$  and  $d_G(y, z) < 2^{m_0}$ . By the triangle inequality,  $d_G(x, z) < 2^{m_0+1}$ . Since  $D \cap \{2^{m_0}, \dots, 2^{m_0+k}\} = \emptyset$ ,  $d_G(x, z) < 2^{m_0}$ , and therefore  $(x, z) \in R$ .  $\square$

For every equivalence class  $U \in T/R$ , we pick an arbitrary  $u \in U$ , and define  $\hat{U} = B_G(u, 2^{m_0})$ .

**Claim 4.2.**  *$\mathcal{U} := \{\hat{U}\}_{U \in T/R}$  is a partial partition of  $V$ . Moreover, for every  $U \in T/R$ ,  $U \subseteq \hat{U}$ ,  $G[\hat{U}]$  is connected and of diameter at most  $2^{m_0+1} < 2^{k^3}$ .*

*Proof.* Let  $U \in T/R$ . Let  $u \in U$  be such that  $\hat{U} = B_G(u, 2^{m_0})$ . For every  $x \in U$ , by the definition of  $R$ ,  $d(x, u) < 2^{m_0}$ , and thus  $x \in \hat{U}$ . Therefore  $U \subseteq \hat{U}$ . By the definition of a ball,  $G[\hat{U}]$  is connected and of diameter at most  $2^{m_0+1} < 2^{k^3}$ . To see that  $\mathcal{U}$  is a partial partition of  $V$ , take  $U' \in T/R$  such that  $U \neq U'$ , and let  $u' \in U'$  be such that  $\hat{U}' = B_G(u', 2^{m_0})$ . Since  $(u, u') \notin R$ ,  $d_G(u, u') \geq 2^{m_0}$ , and since  $D \cap \{2^{m_0}, \dots, 2^{m_0+k}\} = \emptyset$ ,  $d_G(u, u') \geq 2^{m_0+k+1}$ , thus  $\hat{U} \cap \hat{U}' = \emptyset$ .  $\square$

## 4.1 Detailed Algorithm

**Input:**  $G = (V, E, w)$ ,  $T = \{t_1, \dots, t_k\} \subseteq V$   
**Output:** A partition  $\{V_1, V_2, \dots, V_k\}$  of  $V$ .

- 1: rescale the edge weights so that the minimal terminal distance is 1.
- 2: **if**  $\mathcal{D} := \max_{u,v \in T} d_G(u, v) \leq 2^{k^3}$  **then**
- 3: run Algorithm 1, and return its output.
- 4: **else**
- 5: define  $R$  and  $\mathcal{U}$  as above.
- 6: **for all**  $\hat{U} \in \mathcal{U}$  **do**
- 7: run Algorithm 1 independently on  $G[\hat{U}]$ .
- 8: contract  $\hat{U}$  to a single “super-terminal”, maintaining edge weights of all remaining edges.
- 9: denote the resulting graph  $\tilde{G}$ .
- 10: run Algorithm 2 recursively on  $\tilde{G}$  with the set of super-terminals.
- 11: **for all** super-terminals  $u \in \tilde{G}$  **do**
- 12: **for all** vertices  $v$  assigned to  $u$  in the recursive call **do**
- 13: assign  $v$  to its nearest terminal among the nodes contracted to  $u$  in line 8.
- 14: **return** the resulting partition of  $V$ .

**Algorithm 2:** Partitioning  $V$  - The General Case

It is clear that Algorithm 2 returns a partition of  $V$ . In addition, since every level of recursion decreases the number of terminals in the graph, the depth of the recursion is at most  $k$ . During each level of the recursion, Algorithm 1 is invoked at most  $k$  times. Therefore, Algorithm 1 is invoked at most  $k^2$  times, each time on a set of at most  $k$  terminals. By Remark 1, setting  $d = 5$ , we get that there exists  $C_0 > 0$ , such that every time the algorithm is invoked, it achieves a weight stretch factor of at most  $C_0 \log^6 k$ , with probability at least  $1 - O(k^{-3})$ . Applying a union bound, we get that with high probability, the stretch bound is obtained in all invocations of the algorithm. It remains to show that this suffices to achieve the desired stretch factor in  $G$ .

**Lemma 4.3.** *With probability at least  $1 - 1/k$ , on a graph with  $\tau \leq k$  terminals, the algorithm obtains a stretch factor of at most  $C_0 \log^6 k + \log^6 k \cdot 2^{-k} \sum_{k' \leq \tau} 2(k')^2 \leq 2C_0 \log^6 k$ .*

*Proof.* It is enough to show that conditioned on the event that every invocation of Algorithm 1 achieves a stretch factor of at most  $C_0 \log^6 k$ , the generalized algorithm achieves the desired stretch factor. We prove this by induction on  $k$ . For  $k = 2$  the result is clear. Assuming correctness for every  $\tilde{k} < k$ , we prove correctness for  $k$ . Let  $s, t \in T$ . If  $(s, t) \in R$ , then  $s$  and  $t$  are in the same set in  $\mathcal{U}$ , and by the conditioning, the stretch factor of the distance between  $s$  and  $t$  is at most  $C_0 \log^6 k$ . This does not change in steps 4 – 7 of the algorithm. Otherwise, Let  $\tilde{s}, \tilde{t}$  be the terminals (or super-terminals) associated with  $s$  and  $t$  in  $\tilde{G}$  respectively. Denote  $d = d_{G,w}(s, t)$  and  $\tilde{d} = d_{\tilde{G},\tilde{w}}(\tilde{s}, \tilde{t})$ . Denote by  $G' = (V', E', w')$  the terminal-centered minor induced by the partition returned by the recursive call, and by  $G'' = (V'', E'', w'')$  the terminal-centered minor induced by the partition returned in the final step of the algorithm. Denote  $d' = d_{G',w'}(\tilde{s}, \tilde{t})$ . and  $d'' = d_{G'',w''}(s, t)$ . Let  $u$  be a super terminal on a shortest path  $P'$  between  $\tilde{s}$  and  $\tilde{t}$  in  $G'$ . Let  $P''$  be the path obtained from  $P'$  in  $G''$  in the following manner. In the place of every super-terminal  $u$  in  $P'$ , originating in some node set  $\hat{U}$ , we add a path between the corresponding terminals in  $\hat{U}$  (based on the terminal-centered

minor constructed for  $G[\hat{U}]$  in step 3). The edges of  $P'$  are also replaced with corresponding edges in  $G''$ .

Recall that in  $G'$ , the weight of every edge is the distance between its endpoints in  $\tilde{G}$  (by the definition of a terminal-centered minor). In  $G''$  the weight of every edge is the distance between its endpoints in  $G$ . Therefore the weight  $P'$  contained at most  $k - 1$  edges. In  $G''$  the weight of each such edge increases by at most  $k2^{m_0}$ . In addition, every expansion of a super-terminal adds at most  $k2^{m_0}$  to the path. Therefore,  $w''(P'') \leq w'(P') + 2k^2 2^{m_0} \leq d' + d \cdot 2k^2 2^{-k}$ . By the induction hypothesis

$$d' \leq \left( C_0 \log^6 k + \log^6 k \cdot 2^{-k} \sum_{k' \leq k-1} (k')^2 \right) \tilde{d}.$$

Since  $\tilde{d} \leq d$ , we get that

$$\begin{aligned} d'' &\leq \left( C_0 \log^6 k + \log^6 k \cdot 2^{-k} \sum_{k' \leq k-1} 2(k')^2 \right) d + d \cdot 2k^2 2^{-k} \\ &\leq \left( C_0 \log^6 k + \log^6 k \cdot 2^{-k} \sum_{k' \leq k} 2(k')^2 \right) d. \end{aligned}$$

□

## References

- [AGMW10] I. Abraham, C. Gavoille, D. Malkhi, and U. Wieder. Strong-diameter decompositions of minor free graphs. *Theor. Comp. Sys.*, 47(4):837–855, November 2010.
- [Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science*, pages 184–193. IEEE, 1996.
- [Bar04] Y. Bartal. Graph decomposition lemmas and their role in metric embedding methods. In *12th Annual European Symposium on Algorithms*, volume 3221 of *LNCS*, pages 89–97. Springer, 2004.
- [BG08] A. Basu and A. Gupta. Steiner point removal in graph metrics. Unpublished Manuscript, available from <http://www.math.ucdavis.edu/~abasu/papers/SPR.pdf>, 2008.
- [BK96] A. A. Benczúr and D. R. Karger. Approximating s-t minimum cuts in  $\tilde{O}(n^2)$  time. In *28th Annual ACM Symposium on Theory of Computing*, pages 47–55. ACM, 1996.
- [BSS09] J. D. Batson, D. A. Spielman, and N. Srivastava. Twice-ramanujan sparsifiers. In *41st Annual ACM symposium on Theory of computing*, pages 255–262. ACM, 2009.
- [Chu12] J. Chuzhoy. On vertex sparsifiers with Steiner nodes. In *44th symposium on Theory of Computing*, pages 673–688. ACM, 2012.

- [CKR01] G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 8–16. SIAM, 2001.
- [CLLM10] M. Charikar, T. Leighton, S. Li, and A. Moitra. Vertex sparsifiers and abstract rounding algorithms. In *51st Annual Symposium on Foundations of Computer Science*, pages 265–274. IEEE Computer Society, 2010.
- [CXKR06] T. Chan, D. Xia, G. Konjevod, and A. Richa. A tight lower bound for the Steiner point removal problem on trees. In *9th International Workshop on Approximation, Randomization, and Combinatorial Optimization*, volume 4110 of *Lecture Notes in Computer Science*, pages 70–81. Springer, 2006.
- [EGK<sup>+</sup>10] M. Englert, A. Gupta, R. Krauthgamer, H. Räcke, I. Talgam-Cohen, and K. Talwar. Vertex sparsifiers: New results from old techniques. In *13th International Workshop on Approximation, Randomization, and Combinatorial Optimization*, volume 6302 of *Lecture Notes in Computer Science*, pages 152–165. Springer, 2010.
- [FRT04] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004.
- [GNR10] A. Gupta, V. Nagarajan, and R. Ravi. Improved approximation algorithms for requirement cut. *Operations Research Letters*, 38(4):322–325, 2010.
- [Gup01] A. Gupta. Steiner points in tree metrics don’t (really) help. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 220–227. SIAM, 2001.
- [KR11] R. Krauthgamer and T. Roughgarden. Metric clustering via consistent labeling. *Theory of Computing*, 7(5):49–74, 2011.
- [KZ12] R. Krauthgamer and T. Zondiner. Preserving terminal distances using minors. In *39th International Colloquium on Automata, Languages, and Programming*, volume 7391 of *Lecture Notes in Computer Science*, pages 594–605. Springer, 2012.
- [LM10] F. T. Leighton and A. Moitra. Extensions and limits to vertex sparsification. In *42nd ACM symposium on Theory of computing*, STOC, pages 47–56. ACM, 2010.
- [LN05] J. R. Lee and A. Naor. Extending lipschitz functions via random metric partitions. *Inventiones Mathematicae*, 1:59–95, 2005.
- [MM10] K. Makarychev and Y. Makarychev. Metric extension operators, vertex sparsifiers and lipschitz extendability. In *51st Annual Symposium on Foundations of Computer Science*, pages 255–264. IEEE, 2010.
- [MN07] M. Mendel and A. Naor. Ramsey partitions and proximity data structures. *J. Eur. Math. Soc.*, 9(2):253–275, 2007.
- [Moi09] A. Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *50th Annual Symposium on Foundations of Computer Science*, FOCS, pages 3–12. IEEE, 2009.

- [PS89] D. Peleg and A. A. Schäffer. Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989.
- [TZ05] M. Thorup and U. Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005.