SIAM J. COMPUT.
Vol. 48, No. 2, pp. 227–248

USING PETAL-DECOMPOSITIONS TO BUILD A LOW STRETCH SPANNING TREE*

ITTAI ABRAHAM† AND OFER NEIMAN‡

Abstract. We prove that any weighted graph $G = (V, E, w)$ with n points and m edges has a spanning tree T such that $\sum_{\{u,v\}\in E} \frac{d_T(u,v)}{w(u,v)} = O(m \log n \log\log n)$. Moreover, such a tree can be found in time $O(m \log n \log\log n)$. Our result is obtained using our new *petal-decomposition* approach which guarantees that the radius of each cluster in the tree is at most four times the radius of the induced subgraph of the cluster in the original graph.

Key words. low stretch, spanning tree, distortion, embedding

AMS subject classifications. 68R10, 05C85

DOI. 10.1137/17M1115575

1. Introduction.

Let $G = (V, E, w)$ be a finite graph, where $w : E \to \mathbb{R}_+$ is a weight function on the edges. For any subgraph $H = (V', E', w')$ of G, let d_H be the induced shortest path metric with respect to H, where w' is the restriction of w to E'. In particular, for any edge $\{u, v\} \in E$ and any spanning tree T of G, $d_T(u, v)$ denotes the (unique) shortest path distance between u and v in T.

Given a spanning tree T, let

$$\text{avg stretch}_T(G) = \frac{1}{|E|} \sum_{\{u,v\}\in E} \frac{d_T(u,v)}{w(u,v)}. \tag{1}$$

Let avg stretch$(n) = \max_{G=(V,E,w)\,:\,|V|=n} \min_T\{\text{avg stretch}_T(G)\}$. Figure 1 summarizes current progress on the bounds for avg stretch(n) and the time complexity of building such trees.

	avg stretch(n)	Time
[AKPW95]	$\Omega(\log n)$, $\exp(O(\sqrt{\log n \log\log n}))$	$O(m^2)$
[EEST05]	$O((\log n)^2 \log\log n)$	$O(m \log^2 n)$
[ABN08]	$O(\log n(\log\log n)^3)$	$O(m \log^2 n)$
[ABN08]	$O(\log n \log\log n(\log\log\log n)^3)$	$O(m^2)$
[KMP11]	$O(\log n(\log\log n)^3)$	$O(m \log n \log\log n)$
This paper	$O(\log n \log\log n)$	$O(m \log n \log\log n)$

FIG. 1. *Summary of progress on a low stretch spanning tree.*

For the class of Series-Parallel graphs, Emek and Peleg [EP06] obtained avg stretch$(n) = \Theta(\log n)$.

*Received by the editors February 9, 2017; accepted for publication (in revised form) January 7, 2019; published electronically March 21, 2019. A preliminary version of this paper appeared in STOC 2012.
http://www.siam.org/journals/sicomp/48-2/M111557.html
Funding: The second author was supported in part by ISF grant 1817/17 and BSF grant 2015813.
†VMware, Palo Alto, CA 94043 (iabraham@vmware.com).
‡Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, 84105, Israel (neimano@cs.bgu.ac.il).

The main result of this paper is a new upper bound on avg stretch(n) that is tight up to an $O(\log\log n)$ factor and can be constructed in time $O(m\log n\log\log n)$.

THEOREM 1.

$$\text{avg stretch}(n) = O\left(\log n \log\log n\right).$$

Moreover, such a tree can be found in $O(m\log n\log\log n)$ time.

Our tree is created by a hierarchical clustering method we call *petal-decomposition*, which has the desirable property that the radius of the tree created from each cluster is at most four times larger than the radius of the cluster. This technique opens a new path towards obtaining optimal $O(\log n)$ average stretch. Due to technical difficulties described below, the current optimal padded partitions [FRT03, Bar04] cannot be used with the petal-decomposition, so we applied the suboptimal partitions of [Sey95, EEST05]. It remains an open question whether one can construct a suitable optimal strong diameter padded partition, which would imply $O(\log n)$ average stretch. See section 1.2.2 for more details.

Since our result holds for multigraphs, it follows from a reduction of [EEST05] that given any probability distribution over pairs $c : \binom{V}{2} \to [0,1]$, there exists a spanning tree T with

$$\sum_{\{u,v\}\in\binom{V}{2}} c(u,v)\cdot\frac{d_T(u,v)}{w(u,v)} = O(\log n\log\log n)\ .$$

By the minimax theorem (see, e.g., [AF09]), there exists a distribution over spanning trees such that for any edge $\{u,v\}\in E$,

$$\mathbb{E}\left[\frac{d_T(u,v)}{w(u,v)}\right] \le O(\log n\log\log n)\ .$$

There are several algorithmic applications for low stretch spanning trees, such as minimum cost communication spanning trees [Hu74] and network visualization [McK15]; we refer the reader to [EEST05, ABN08] for more details. One of the applications is a fast solver for a symmetric diagonally dominant system of linear equations [ST04, KOSZ13, CKM$^+$14]. There are very recent works (e.g., [KS16, KLP$^+$16]) exhibiting faster solvers that do not use low stretch spanning trees.

1.1. Related work. Embedding metric spaces and graphs into tree metrics and spanning trees has received a lot of attention over the last two decades. The basic motivation is that problems on simple graphs such as trees are often much easier than those on arbitrary graphs, and embedding the original graph into a tree (or a distribution over trees) is a basic step in approximation algorithms, network design, online algorithms, and other settings. As mentioned above, the first results were obtained by [AKPW95], who showed an $\exp(O(\sqrt{\log n\log\log n}))$ bound on the average stretch. If we drop the requirement that the tree is spanning (that is, allow adding and not only deleting edges, while maintaining that distances in the tree are no smaller than those in the graph), then [Bar96, Bar98, CCGG98, FRT03] in a sequence of works showed optimal average stretch of $\Theta(\log n)$. This line of work proved very fruitful, because in many settings we can suffer from nonspanning trees. If we replace the right-hand side of (1) by averaging over all pairs, then [ABN07] showed a universal constant bound on that quantity, which is called the average distortion.

A related line of research studies a *relative guarantee* approximation: Given a graph, can we approximate the best possible tree. For the question of maximum stretch over all pair distances, [BIS07] showed how to obtain a $(c \log n)^{O(\sqrt{\log \Delta})}$ factor, where c is the optimal maximum stretch and Δ is the diameter. They also showed $O(1)$ approximation for the case where the graph is unweighted. The constant was recently improved by [CDN$^+$10]. For embedding unweighted graphs into a *spanning* tree, [EP04] showed $O(\log n)$ approximation for maximum stretch. However, for the setting of average stretch, essentially nothing is known (except for the trivial $\tilde{O}(\log n)$[1] absolute bound shown here and in [ABN08]).

1.2. Techniques.

1.2.1. Petal decomposition and radius increase. The star-decomposition technique of Elkin et al. [EEST05] is a method to iteratively build a spanning tree. In each iteration it partitions the vertices of the current graph into clusters that are connected in a star structure: a central cluster is connected to every other cluster by a single edge, and all other edges between clusters are dropped. In both previous manifestations of star-decompositions (see [EEST05] and [ABN08]) the first step in each iteration is to define the central cluster as an appropriately chosen ball around some center point. After the central ball is defined, then the remaining clusters (called cones) are defined sequentially.

The radius of a graph is the maximal distance from a designated center. One of the main difficulties in the spanning tree construction is that the radius may increase by a small factor at every application of the star decomposition, which translates to increased stretch. If we drop the requirement that the tree is a *spanning tree* of the graph and just require a tree metric, then this difficulty does not appear, and indeed optimal $\Theta(\log n)$ bound is known on the average stretch [FRT03, Bar04]. In order to control the radius increase, [EEST05] had to pay an additional factor of $O(\log n)$. This was improved by [ABN08], where a subtle change to the algorithm and a careful analysis of the radius increase allowed the factor to be reduced to $\tilde{O}(\log \log n)$. One of the main contributions of this work is a new decomposition scheme which we call *petal-decomposition*, allowing essentially optimal control on the radius increase of the spanning tree; it increases by at most a factor of 4 over all the recursion levels.

Our new *petal-decomposition* technique is also a method to iteratively build a spanning tree. In each iteration it starts by sequentially building a series of clusters which we call *petals*. Once no more petals can be built, the remaining central cluster is called the *stigma*. Then the petals and the stigma are connected to form a tree using some of the intercluster edges, and all other edges between clusters are dropped.

The petal-decomposition approach differs from star-decompositions in several aspects.

1. First, it is not the case that all petals are necessarily connected to the stigma by an edge (as would be the case in the star-partition); each petal is connected either to the stigma or to another petal by an edge. The petal connections form a rooted tree whose root is the stigma.
2. Second, the stigma is not necessarily a ball; it is the remaining subgraph once no more petals can be formed.
3. Third, each cone guarantees one edge in order to become part of the tree (the edge connecting the cone to the central ball), while a petal contains a certain shortest-path which will be included entirely in the tree.

In Figure 2, a star-decomposition each cone $C(x_0, x, r)$ is defined by three param-

[1]By $g(n) \leq \tilde{O}(f(n))$ we mean that there exists some constant k such that $g(n) \leq O(f(n)) \cdot \log^k(f(n))$.

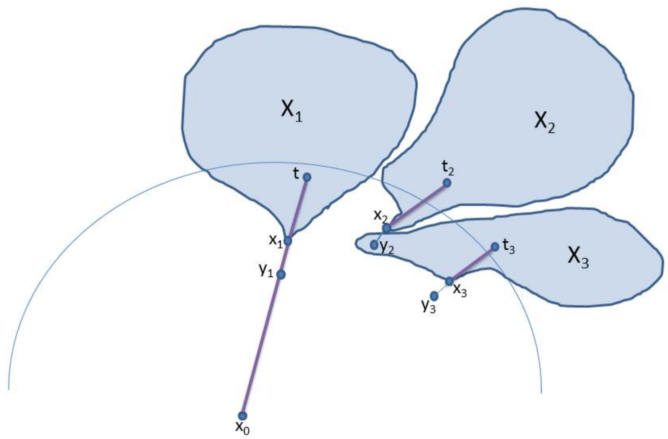

FIG. 2. *Creating the first three petals with their highways. The first portal is connected by a highway to x_0 (this means that the shortest path from x_0 to x_1 will be included in the final tree). Note that the portal edges do not necessarily connect the petal to the stigma, but may connect between petals. In this example, the portal node y_2 of X_2 is contained in the petal X_3. The algorithm guarantees that this cannot happen to the first portal node y_1 (thus y_1 will be a part of the stigma X_0).*

eters: the center of the current cluster x_0, the center of the cone x, and the radius r of the cone; then the cone consists of all the points v such that $d(x_0, x) + d(x, v) - d(x_0, v) \le r$. The radius r of the cone determines the maximum increase in the radius of the graph (with respect to the center x_0).

A petal $P(x_0, t, r)$ is also defined by three parameters: the center of the current cluster x_0, the *target* of the petal t, and the radius r of the petal. The center of the petal (denoted by x) is the point on the shortest path from t to x_0 of distance r from t. Moreover, we call the path from the center of the petal x to the target of the petal t the *highway* of the petal. An important property of our construction is that this highway path is guaranteed to be a part of the final spanning tree, which is achieved by the so-called *special petal*. The special petal is defined as a union of cones of varying radii. Specifically, let p_k be the point of distance k from the target t on the shortest path from t to x_0. Then the petal $P(x_0, t, r)$ is defined as the union of cones $C(x_0, p_k, (r - k)/2)$ for all $k \le r$.

Informally, the crucial property of a petal and its highway is the following: Assume $z \in P(x_0, t, r)$, and that $P_{x_0 z}$ is the shortest path from the center x_0 to z. By forming the petal, we remove all edges between $P(x_0, t, r)$ and $G \setminus P(x_0, t, r)$ except for the edge from the petal center x towards the center of the current cluster. Hence every path from x_0 to z will go through the petal center x. If the new shortest path $P'_{x_0 z}$ (after forming the petal) is (additively) α longer than the length of $P_{x_0 z}$, then $P'_{x_0 z}$ will contain part of the highway of length at least 2α; see Figure 3. Such a property could foster wishful thinking: Suppose that in each iteration we increase the distance of a point to the center by at most α, but also mark a new portion of the path of length 2α as edges that are guaranteed to appear in the final tree (part of a highway). In such a case it is easy to see that the final path will have stretch at most 2 (if the original distance was b, once the total increase is b we have marked $2b$—all of the path—as a highway that will appear in the tree). Unfortunately, the shortest path

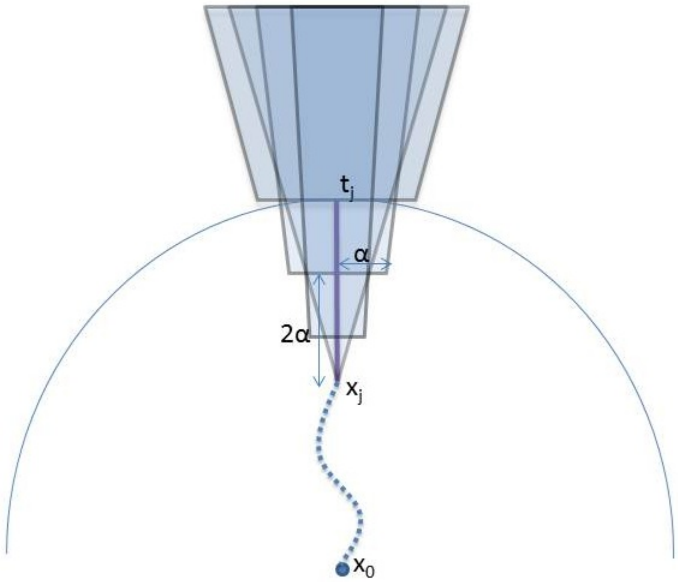

FIG. 3. *Definition of a petal with target t_j, center x_j, and highway path $P_{x_j t_j}$. The side radius of each cone (that determines the maximum increase in the radius) is half of the highway path.*

from x to z in the final tree may not use the prescribed highway of the parent cluster. There may be a shorter path that avoids this highway, so the above “wishful thinking” argument does not work.

The key algorithmic idea to alleviate this problem is to decrease the weight of an edge by half when it becomes part of a highway (we ensure that this happens at most once for every edge). This reweighting signals later iterations to use the prescribed highway (or use a shorter path, no longer than the reweighted highway). Therefore, if we generate a *new* highway in the path from x_0 to some z when we form $P(x_0, t, r)$, then (after reweighting the highway) *the length of the path does not increase at all* (it increased by at most α, but length of at least 2α was reduced by 1/2). The other case is that no new highway is generated, which can only happen for the first cluster created (some of the highway edges may have been reweighted already). In this case we turn to the idea of [ABN08], that one may choose a certain *target* point y_1 and have that the shortest path connecting x_0 to y_1 will appear in the tree. Here we choose y_1 as the point leading to the first cluster. This approach implies that even though we may increase the radius, a constant fraction of the path is guaranteed not to increase ever again. We use a subtle inductive argument to make this intuition precise, and in fact we lose a factor of 2 for each of these cases, so the maximal increase is by a factor of 4.

Constructing petals. An alternative way to define cones $C(x_0, x, r)$ and petals $P(x_0, t, r)$ as a ball growing procedure on a directed graph shows their similarities. This view is essential for a fast algorithm to construct the petals. Given a weighted undirected graph $G = (V, E)$ with a center x and a target t, let p_k be the point of distance k from t on the shortest path P_{tx} from t to x; all distances d are with respect to G. Let $\tilde{G} = (V, A, \bar{w})$ be the weighted directed graph induced by adding the two directed edges $(u, v), (v, u) \in A$ for each $\{u, v\} \in E$ and setting $\bar{w}(u, v) = w(u, v) - (d(v, x) - d(u, x))$. The cone $C(x, t, r)$ is simply the ball around t of radius

r in $\tilde{G}$. The Petal $P(x,t,r)$ is the ball around t of radius $r/2$ in $\tilde{G}$ with one change: the weight of each edge on the path $P_{t,p_{r'}}$ is changed to be $1/2$ of its original weight (that is, $1/2$ of its weight in G), where $r' \le r$ is the maximal such that there exists a point $p_{r'}$ on this path.

We shall elaborate on this in section 6.

1.2.2. Sparse graph decompositions. A basic tool that is often used in constructing tree metrics and spanning trees with low stretch is sparse graph decomposition. The idea is to partition the graph into small diameter pieces such that few edges are cut. Each cluster of the decomposition is partitioned recursively, which yields a hierarchical decomposition. Creating a tree recursively on each cluster of the decomposition, and connecting these in a tree structure, will yield a spanning tree of the graph. The edges cut by the decomposition are potentially stretched by a factor proportional to the diameter of the created tree. The construction has to balance these two goals: cut a small number of edges while maintaining small diameter in the created tree.

For a spanning tree we require both *strong diameter* partitions and control of the diameter increase. The authors of [EEST05] build a tree with average stretch $O(\log^2 n \log\log n)$. A factor of $O(\log n \log\log n)$ is due to the partitions based on the approach of [Sey95, Bar98], and another $O(\log n)$ is required to control the diameter of the tree. The publication [ABN08] has a factor of $O(\log n)$ due to the partitions based on the approach of [Bar04, ABN06], and another $\tilde{O}(\log\log n)$ is required to control the diameter of the tree.

In this work, we show a new petal-decomposition that incurs only a constant cost to control the diameter of the tree. We hoped that the partition cost would be based on local growth ratio bounds (as in [FRT03, Bar04, ABN06, ABN08]), which would lead to optimal average stretch. Known strong diameter partitions (see [ABN08]) that obtain a local growth ratio bound require a careful selection of the center of each cluster. However, our current petal-decomposition approach does not allow one to choose the centers arbitrarily, and hence we could not directly use the technique of [ABN08]. Therefore, we turn to the partitions of [EEST05], which is the only reason for the extra $O(\log\log n)$ factor. It remains an open question whether one can construct an optimal strong diameter partition whose centers can be chosen arbitrarily. Our results show that this open question is the only barrier to obtaining an optimal low stretch tree.

Subsequent work. In [EFN15] the notion of *terminal embedding* was introduced. Given a graph $G = (V, E)$ with a designated set of terminals $K \subseteq V$, they want an embedding that approximately preserves distances for all pairs containing a terminal, whose distortion depends only on $k = |K|$. Using our petal-decomposition they devise a distribution over spanning trees with expected stretch $\tilde{O}(\log k)$ (for any pair containing a terminal).

In [ACE$^+$18] the petal-decomposition approach was used to create *Ramsey spanning trees*. This can be viewed as a small collection of spanning trees, where each vertex v has a tree in the collection that provides small stretch to all pairs containing v.

1.3. Structure of the paper. In section 3 we describe the new petal-decomposition and prove some of its basic properties. In section 4 we bound the total radius increase by a factor of 4. In section 5 we analyze the total stretch, and provide the improved bound of $O(\log n \log\log n)$ on the average stretch, proving the first statement of Theorem 1. In section 6 we show an alternative view of forming a petal by shortest paths in a certain graph that concludes the proof of Theorem 1.

2. Preliminaries. Let $G = (V, E, w)$ be a weighted undirected graph; we shall assume throughout the paper that $w(e) \geq 1$ for all edges $e \in E$. For any $X \subseteq V$, $G(X)$ is the subgraph induced on X with edges $E(X) = \{\{u, v\} \in E \mid u, v \in X\}$. For $Y \subseteq X$ denote by $\partial_X(Y) = \{\{u, v\} \in E(X) \ : \ |Y \cap \{u, v\}| = 1\}$ the set of edges with exactly one end point in Y (in the graph induced by X). Let $d_X : X^2 \to \mathbb{R}^+$ be the shortest path metric in $G(X)$. Let $\mathrm{diam}(X) = \max_{y,z \in X}\{d_X(y, z)\}$. For $x \in X$ let $\mathrm{rad}_x(X) = \max_{y \in X}\{d_X(x, y)\}$; we omit the subscript when it is clear from the context (note that $\mathrm{diam}(X)/2 \leq \mathrm{rad}(X) \leq \mathrm{diam}(X)$). For any $x \in X$ and $r \geq 0$ let $B_{(X,d)}(x, r) = \{y \in X \mid d_X(x, y) \leq r\}$ (we omit the metric d whenever $d = d_X$ is the standard shortest-path metric).

By *contracting* an edge $\{u, v\}$ we mean identifying its endpoints to a single vertex w, while preserving all the edges leaving u or v (so we may have multiple edges leaving w). By *expanding back* w we undo the contracting operation.

For a spanning tree $T = T[X]$ of a subgraph $G(X)$, define the total stretch on edges in $G(X)$ as

$$\mathrm{TS}[X] = \sum_{\{u,v\} \in E(X)} \frac{d_T(u, v)}{w(u, v)} \ .$$

For $X \subseteq V$ and vertices $u, v \in X$, let $P_{uv}(X)$ be a fixed shortest path between u, v in $G(X)$ (assuming that $G(X)$ is connected). We may assume that there is a unique such path. By $T = \mathrm{SPT}_x(G(X))$ we mean the shortest path tree rooted at x (the subscript is dropped when the center x is clear from the context). Such a tree may be constructed in near linear time using Dijkstra's algorithm, which satisfies $d_T(x, y) = d_X(x, y)$ for all $y \in X$.

A *ξ-concentric system* $\{L_r \subseteq X \ : \ r \geq 0\}$ is a generalization of balls and should satisfy that $L_0 \neq \emptyset$, $L_r \subseteq L_{r'}$ for $r < r'$, and for an edge $\{u, v\}$, if $u \in L_r$, then $v \in L_{r+\xi \cdot w(u,v)}$. For balls we have $\xi = 1$, and for petals defined below we will have $\xi = 2$.

Define the *cost* of an edge $e \in E$ to be $\mathrm{cost}(e) = 1/w(e)$, and the cost of a set of edges $F \subseteq E$ is $\mathrm{cost}(F) = \sum_{e \in F} \mathrm{cost}(e)$. For a 2-concentric system $\{L_r\}$ we would like to define the *volume* of its members as the number of edges they contain plus the appropriate fraction of the edges in $\partial_X(L_r)$. To make this precise, fix $r \geq 0$ and $e = \{u, v\}$, and let r_u (resp., r_v) be the minimal r such that $u \in L_r$ (resp., $v \in L_r$). Without loss of generality, we assume $r_u \leq r_v$, and define for $r \in [r_u, r_v]$, $\alpha_e = \frac{r - r_u}{2w(u,v)}$. Intuitively, α_e measures the "fraction" of e that lies in L_r. Observe also that since $\{L_r\}$ is a 2-concentric system, we have $r_v \leq r_u + 2w(u, v)$, so that $\alpha_e \in [0, 1]$, and

$$\mathrm{vol}_X(L_r) = 1 + |E(L_r)| + \sum_{e \in \partial_X(L_r)} \alpha_e \ .$$

The 1 is added for technical convenience. We shall omit the subscript X where X is clear from the context. Observe that the volume is a nondecreasing function of the radius r, and the derivative of this function is half the cost of the edges in $\partial_X(L_r)$ (as long as there is no edge leaving $\partial_X(L_r)$ when we increase r slightly).

DEFINITION 1 (cone-metric[2]). *Given a graph $G = (V, E, w)$, subset $X \subseteq V$, and*

[2]In fact, the cone-metric is a pseudo-metric.

$T = $ `hierarchical-petal-decomposition`$(G(X), x_0, t)$:
1. If $|X| = 1$, return T consisting of a single vertex.
2. $(X_0, \ldots, X_s, (y_1, x_1), \ldots, (y_s, x_s), t_0, \ldots, t_s)$
   $=$ `petal-decomposition`$(G(X), x_0, t)$;
3. For each $j \in [0, \ldots, s]$:
   (a) $T_j = $ `hierarchical-petal-decomposition`$(G(X_j), x_j, t_j)$;
4. Let T be the tree formed by connecting $T_0, \ldots, T_s$ using the edges $\{y_1, x_1\}, \ldots, \{y_s, x_s\}$;

FIG. 4. *hierarchical-petal-decomposition algorithm. The input is a graph $G(X)$ and two vertices $x_0, t \in X$, and the output is a spanning tree of $G(X)$.*

points $x, y \in X$, define the cone-metric $\rho = \rho(X, x, y) : X^2 \to \mathbb{R}^+$ *as* $\rho(u, v) = |(d_X(x, u) - d_X(y, u)) - (d_X(x, v) - d_X(y, v))|$.

Observe that this definition is slightly different from the definition given in [ABN08], which is based on [EEST05] (the one given here is less general, but suffices for our needs). Note that a ball $B_{(X,\rho)}(y, r)$ in the cone-metric $\rho = \rho(X, x, y)$ is the set of all points $z \in X$ such that $d_X(x, y) + d_X(y, z) - d_X(x, z) \le r$. Note that distances in the cone-metric can be at most twice the shortest-path distance (and that they can be much shorter); this is because $\rho(u, v) \le |d_X(u, x) - d_X(x, v)| + |d_X(v, y) - d_X(y, u)| \le 2d_X(u, v)$ (where X, x, y are as defined above). This implies that for any $z \in V$, $\{B_{(X,\rho)}(z, r)\}_r$ is a 2-concentric system, since for $u \in B_{(X,\rho)}(z, r)$ and an edge $\{u, v\} \in E$, we see that $v \in B_{(X,\rho)}(z, r + 2d_X(u, v))$ (and $d_X(u, v) \le w(u, v)$). In particular, for a ball B in a cone-metric

$$\mathrm{vol}_X(B) \le 1 + |E(B)| + 2|\partial(B)| \ . \tag{2}$$

3. Petal-decomposition.

Hierarchical-petal-decomposition algorithm. See Figure 4 for the algorithm. Let $G = (V, E)$ be a weighted graph. Here and in what follows $n = |V|$ and $m = |E|$. For ease of presentation we first assume that the aspect ratio of the graph (the ratio between the largest and smallest distance) is at most polynomial in n, and in section 6 we remark how to handle arbitrary weights. Create a spanning tree T by choosing an arbitrary vertex $x_0 \in V$ and calling it `hierarchical-petal-decomposition`(G, x_0, x_0). In section 4 we show that this tree satisfies $\mathrm{rad}(T) \le O(\mathrm{rad}(G))$.

3.1. Properties and correctness. Fix some subset $X \subseteq V$, and consider running `petal-decomposition` on $G(X)$ with some $x_0 \in X$ and target $t \in X$. See Figure 5. Denote $\Delta = \mathrm{rad}_{x_0}(X)$. The algorithm partitions X to $X_0, X_1, \ldots, X_s$, finds edges $(x_1, y_1), \ldots, (x_s, y_s)$ connecting these clusters, and assigns centers $t_0, \ldots, t_s$ for each new cluster. The first cluster created X_1 may be a special cluster, whose purpose is to preserve the shortest path from x_0 to t. The main properties we want from this partition are the following:

- Each cluster X_j, $0 \le j \le s$, is connected, has smaller radius $\mathrm{rad}_{x_j}(X_j) \le 3\Delta/4$, and contains the shortest path from its center x_j to its target t_j.
- The edges $(x_1, y_1), \ldots, (x_s, y_s)$ connecting the clusters form a tree (when thinking of each cluster as a single vertex). This is achieved by ensuring $x_j \in X_j$ and $y_j \notin X_1 \cup \cdots \cup X_j$ for each $1 \le j \le s$.
- The shortest path from the center x_0 to the target t is either fully contained in X_0 or in $X_0 \cup X_1$ (if X_1 is special).

$(X_0,\ldots,X_s,(y_1,x_1),\ldots,(y_s,x_s),t_0,\ldots,t_s) =$ `petal-decomposition`$(G(X),x_0,t)$:

1. Let $\Delta = \mathrm{rad}_{x_0}(X)$; Let $Y_0 = X$; Set $j = 1$;
2. If $d_X(x_0,t) > 5\Delta/8$, create a special first petal:
   (a) Let $(X_1,x_1) =$ `create-petal`$(X,X,t,x_0,[d_X(x_0,t) - 5\Delta/8, d_X(x_0,t) - \Delta/2])$;
   (b) $Y_1 = Y_0 \setminus X_1$;
   (c) Let y_1 be the neighbor of x_1 on $P_{x_0 t}$ (the one closer to x_0); Set $t_0 = y_1$, $t_1 = t$;
   (d) Set $j = 2$;
3. Otherwise, if $d_X(x_0,t) \le 5\Delta/8$, set $t_0 = t$.
4. Creating the petals:
   (a) While $Y_{j-1} \setminus B_X(x_0, 3\Delta/4) \neq \emptyset$:
      i. Let $t_j \in Y_{j-1}$ be an arbitrary point satisfying $d_X(x_0,t_j) > 3\Delta/4$;
      ii. Let $(X_j,x_j) =$ `create-petal`$(X,Y_{j-1},t_j,x_0,[0,\Delta/8])$; $Y_j = Y_{j-1} \setminus X_j$ (see Figure 6);
      iii. For each edge $e \in P_{x_j t_j}$, set its weight to be $w(e)/2$;
      iv. Let y_j be the neighbor of x_j on $P_{x_0 t_j}$ (the one closer to x_0);
      v. Let $j = j + 1$;
   (b) Let $s = j - 1$;
5. Creating the stigma X_0:
   (a) Let $X_0 = Y_s$;

FIG. 5. *`petal-decomposition` algorithm. The input is a graph $G(X)$ and two vertices $x_0, t \in X$. The output is a partition of X into connected clusters $X_0,\ldots,X_s$ of radius at most $3\mathrm{rad}(X)/4$, and s edges $(x_1,y_1),\ldots,(x_s,y_s)$ connecting these clusters. These edges satisfy $x_j \in X_j$ for all $0 \le j \le s$, and $y_j \notin X_1 \cup \cdots \cup X_j$ (i.e., y_j is not clustered by the first j clusters). Finally, targets $t_0,\ldots,t_s$ with $t_j \in X_j$ are specified for each new cluster.*

$(W,x) =$ `create-petal`$(X,Y,t,x_0,[lo,hi])$:

1. Let $R = hi - lo$;
2. Define $W_r = \bigcup_{p \in P_{x_0 t}\ :\ d_Y(p,t) \le r} B_{(Y,\rho(Y,x_0,p))}(p,(r - d_Y(p,t))/2)$;
3. Let $L = \lceil \log\log n \rceil$; Set $q = 1$;
4. Loop:
   (a) Set $a = lo + (q-1)R/L$; Fix $\chi = \frac{|E(X)|+1}{\mathrm{vol}_Y(W_a)}$;
   (b) If $\mathrm{vol}_Y(W_a) \le \frac{4|E(X)|}{2^{(\log m)^{1-q/L}}} + 1$, exit the loop; Otherwise, set $q = q+1$ and repeat;
5. Find $r \in [a, a + R/L]$ such that $\mathrm{cost}(\partial_Y(W_r)) \le \mathrm{vol}_Y(W_r) \cdot \frac{2L \ln \chi}{R}$;
6. Let $r' \le r$ be the maximal such that there exists a point $p_{r'}$ with $d_Y(t,p_{r'}) = r'$ on $P_{x_0 t}$;
7. Return $(W_r, p_{r'})$;

FIG. 6. *`create-petal` algorithm. The input is a set of vertices X, a subset $Y \subseteq X$ on which we work right now, two vertices $x_0, t \in Y$, and a range $[lo,hi]$ satisfying $hi - lo = \mathrm{rad}_{x_0}(X)/8$ from which to pick the radius. The output is a cluster $W \subseteq Y$ and a center $x \in W$.*

In what follows we formally prove all of these properties. Recall that $Y_{j-1} = X \setminus (X_1 \cup \cdots \cup X_{j-1})$ is the vertex set of the graph from which a petal is carved in iteration j of the `petal-decomposition` algorithm. First, we show that the shortest path from any $z \in Y_j$ to the center x_0 is fully contained in Y_j. This proof essentially appeared in [EEST05, ABN08], and we include it for the sake of completeness.

Claim 1. *Let $1 \le j \le s$ be an integer, and let $z \in Y_j$; then $P_{x_0 z}(X) \subseteq G(Y_j)$.*

Proof. Seeking a contradiction, assume that $P_{x_0 z}(X) \nsubseteq G(Y_j)$, and let $1 \le h \le j$ be the minimal such that there exists $u \in P_{x_0 z}(X)$ and $u \in X_h$. We will have a contradiction once it is shown that $z \in X_h$ as well. Let x_h and t_h be the center and target of the petal X_h, respectively. Let r_h be the radius that was chosen for creating X_h. Let p_k be the point on $P_{x_h t_h}(Y_{h-1})$ of distance k from t_h such that $u \in B_{(Y_{h-1},\rho)}(p_k, (r_h - k)/2)$, where $\rho = \rho(Y_{h-1}, x_0, p_k)$. By Definition 1 this means that

$$d_{Y_{h-1}}(x_0, u) + (r_h - k)/2 \ge d_{Y_{h-1}}(x_0, p_k) + d_{Y_{h-1}}(p_k, u) \ . \tag{3}$$

By the minimality of h it follows that $P_{x_0 z}(X)$ is fully contained in $G(Y_{h-1})$ (even if $h = 1$, recall that $Y_0 = X$). Since u lies on $P_{x_0 z}(X)$, it follows that $d_{Y_{h-1}}(x_0, z) = d_{Y_{h-1}}(x_0, u) + d_{Y_{h-1}}(u, z)$. Now

$$\begin{aligned} d_{Y_{h-1}}(x_0, z) + (r_h - k)/2 &= d_{Y_{h-1}}(x_0, u) + (r_h - k)/2 + d_{Y_{h-1}}(u, z) \\ &\overset{(3)}{\ge} d_{Y_{h-1}}(x_0, p_k) + d_{Y_{h-1}}(p_k, u) + d_{Y_{h-1}}(u, z) \\ &\ge d_{Y_{h-1}}(x_0, p_k) + d_{Y_{h-1}}(p_k, z); \end{aligned}$$

hence $z \in B_{(Y_{h-1},\rho)}(p_k, (r_h - k)/2)$ and thus should also be in X_h, which yields a contradiction. ◻

Corollary 2. *The cluster X_0 is connected.*

Proof. Applying Claim 1 to $Y_s = X_0$, we conclude that if $z \in X_0$, it is connected to x_0. ◻

Observation 3. *For each $j \ge 1$, $P_{x_j t_j}(X) \subseteq G(X_j)$.*

Proof. As x_j was chosen on the shortest path connecting x_0 to t_j, and since by Claim 1 $P_{x_0 t_j}(X) \subseteq Y_{j-1}$, we get that by definition of cone-metric, $d_{(Y_{j-1},\rho(Y_{j-1},x_0,x_j))}(x_j, p) = 0$ for all $p \in P_{x_j t_j}(X)$. This suggests that the entire path $P_{x_j t_j}(X) \subseteq G(X_j)$. ◻

Corollary 4. *For each integer $j \ge 1$, X_j is connected.*

Proof. By Observation 3, $P_{x_j t_j}(X)$ is fully contained in $G(X_j)$, and since X_j is a union of balls (in a cone-metric) centered at the points of $P_{x_j t_j}(X)$, it is then connected. ◻

The following two claims show that the radii of clusters decreases by a constant factor at each level. They are similar to claims proven in [EEST05, ABN08]; we provide proofs for the sake of completeness.

Claim 5. *$B_X(x_0, \Delta/2) \subseteq X_0 \subseteq B_X(x_0, 3\Delta/4)$.*

Proof. For the right inclusion, note that for any $j \ge 1$, if there is a point in $Y_{j-1} \setminus B_X(x_0, 3\Delta/4)$, we continue creating petals; therefore, $Y_s \setminus B_X(x_0, 3\Delta/4) = \emptyset$ and $X_0 = Y_s \subseteq B_X(x_0, 3\Delta/4)$.

To see the left inclusion, fix any $z \in X$ with $d_X(x_0, z) \le \Delta/2$; we will show that $z \notin X_j$ for any $j \ge 1$. First, consider the case when the special petal is built, which happens when $d_X(x_0, t) > 5\Delta/8$, and we conclude that the petal radius r_1 is at most

$$r_1 \le hi = d_X(x_0, t) - \Delta/2 \ . \tag{4}$$

For any $0 \le k \le r_1$ and $p_k \in P_{x_1 t}(X)$, we now show that the cone of p_k will not contain z. Recall that $d_X(p_k, t) = k$, so that

$$d_X(x_0, p_k) = d_X(x_0, t) - k \ , \tag{5}$$

and we have that

$$\begin{aligned} d_X(z, p_k) + d_X(p_k, x_0) &\ge d_X(p_k, x_0) - d_X(x_0, z) + d_X(p_k, x_0) \\ &\overset{(5)}{\ge} 2d_X(x_0, t) - 2k - \Delta/2 \\ &= (d_X(x_0, t) - \Delta/2 - k)/2 + (3/2)(d_X(x_0, t) - k) - \Delta/4 \\ &\overset{(4)}{>} (r_1 - k)/2 + \Delta/2 \\ &\ge d_X(x_0, z) + (r_1 - k)/2 \ . \end{aligned} \tag{6}$$

By the definition of cone-metric, this implies that $z \notin X_1$. The calculation for the other petals is similar: fix any $j \ge 1$ ($j = 1$ only if there is no special petal), and consider the petal X_j with target t_j with radius $r_j \le \Delta/8$, recalling that $d_X(x_0, t_j) \ge 3\Delta/4$. Here, in fact, we can show a stronger bound; it suffices that $z \in X$ has $d_X(x_0, z) < 5\Delta/8$ to be left outside of the petal X_j. We will use Claim 1 to argue that distances from x_0 in Y_{j-1} are the same as those in X (this also holds for distances between vertices on the shortest-paths emanating from x_0, e.g., on $P_{x_j t_j}(X)$). For any $0 \le k \le r_j$ and $p_k \in P_{x_j t_j}(X)$ we have that

$$\begin{aligned} d_{Y_{j-1}}(z, p_k) + d_{Y_{j-1}}(p_k, x_0) &\ge d_X(p_k, x_0) - d_X(x_0, z) + d_X(p_k, x_0) \\ &\ge 2d_X(x_0, t_j) - 2k - 5\Delta/8 \\ &\ge (3/2)\Delta - k/2 - (3/2) \cdot \Delta/8 - 5\Delta/8 \\ &= 5\Delta/8 + (\Delta/8 - k)/2 \\ &\ge d_{Y_{j-1}}(x_0, z) + (r_j - k)/2 \ , \end{aligned} \tag{7}$$

where we used that $k \le \Delta/8$ in the third inequality. Again by the definition of cone-metric, this implies that $z \notin X_j$ as well. □

CLAIM 6. *For each* $1 \le j \le s$, $\mathrm{rad}_{x_j}(X_j) \le 3\Delta/4$.

Proof. Consider first the special petal X_1 with radius

$$r_1 \le d_X(x_0, t) - \Delta/2 \ . \tag{8}$$

For $z \in X_1$, let $0 \le k \le r_1$ and $p_k \in P_{x_1 t}(X)$ of distance k from t be such that $z \in B_{(X,\rho)}(p_k, (r_1 - k)/2)$ (with $\rho = \rho(X, x_0, p_k)$). In particular,

$$d_X(x_0, p_k) = d_X(x_0, t) - k \ . \tag{9}$$

By definition of cone-metric we get that

$$d_X(x_0, p_k) + d_X(p_k, z) \le d_X(x_0, z) + (r_1 - k)/2 \ . \tag{10}$$

Using Observation 3 and the fact that each point on $P_{p_k z}(X)$ must be in X_1 as well, recalling $d(x_1, p_k) = r_1 - k$ it follows that

$$\begin{aligned} d_{X_1}(x_1, z) &\leq d_X(x_1, p_k) + d_X(p_k, z) \\ &\overset{(10)}{\leq} r_1 - k + d_X(x_0, z) + (r_1 - k)/2 - d_X(x_0, p_k) \\ &\overset{(9)}{=} 3r_1/2 + d_X(x_0, z) - d_X(x_0, t) - k/2 \\ &\overset{(8)}{\leq} 3(d_X(x_0, t) - \Delta/2)/2 + d_X(x_0, z) - d_X(x_0, t) \\ &\leq 3\Delta/4 \ , \end{aligned}$$

where in the last inequality we use that every distance from x_0 is at most Δ. For the other petals, consider X_j with radius r_j for some $j \geq 1$, and let $z \in X_j$. Let $0 \leq k \leq r_j$ and $p_k \in P_{x_j t_j}(Y_{j-1})$ be of distance k from t such that $z \in B_{(Y_{j-1}, \rho)}(p_k, (r_j - k)/2)$ (with the cone-metric $\rho = \rho(Y_{j-1}, x_0, p_k)$). By definition of cone-metric and as $r_j \leq \Delta/8$, we get that

$$d_{Y_{j-1}}(x_0, p_k) + d_{Y_{j-1}}(p_k, z) \leq d_X(x_0, z) + (r_j - k)/2 < 9\Delta/8 \ . \tag{11}$$

And finally, applying Observation 3 and Claim 1 we get that

$$\begin{aligned} d_{X_j}(x_j, z) &\leq d_{Y_{j-1}}(x_j, p_k) + d_{Y_{j-1}}(p_k, z) \\ &\overset{(11)}{\leq} r_j + 9\Delta/8 - d_X(x_0, p_k) \\ &\leq 5\Delta/4 - 5\Delta/8 \\ &< 3\Delta/4 \ , \end{aligned}$$

where we used in the third inequality that $d(x_0, p_k) \geq 5\Delta/8$, which holds since $r_j \leq \Delta/8$ and the target t_j is at least $3\Delta/4$ away from x_0. ∎

CLAIM 7. *If a special first petal is created, then* $y_1 \in X_0$.

Proof. As the radius r_1 of the special petal is at least $d_X(x_0, t) - 5\Delta/8$, and as $y_1 \notin X_1$, it follows that $d_X(y_1, t) > d_X(x_0, t) - 5\Delta/8$, and as y_1 is on the shortest path $P_{x_0 t}(X)$ we get that $d_X(x_0, y_1) = d_X(x_0, t) - d_X(y_1, t) < 5\Delta/8$. It remains to show that for any other $j \geq 2$, if $z \in X_j$, then $d_X(x_0, z) \geq 5\Delta/8$, but this was proved in Claim 5 (see above, before (7)). ∎

CLAIM 8. *If a special first petal is created, then* $P_{x_0 t}(X) \subseteq G(X_0 \cup X_1)$. *Otherwise,* $P_{x_0 t}(X) \subseteq G(X_0)$.

Proof. If a special petal is created, then it surely contains its target t. Consider the shortest path $P_{x_0 t}(X)$, and divide it into $P_{x_0 y_1}(X)$ and $P_{x_1 t}(X)$. By Observation 3 the path $P_{x_1 t}(X)$ is fully contained in X_1. By Claim 7, $y_1 \in X_0$, and thus Claim 1 implies that $P_{x_0 y_1}(X)$ is also contained in X_0.

If it is the case that there is no special petal, then $d_X(x_0, t) \leq 5\Delta/8$; thus $t \in X_0$, and again by Claim 1 the whole path $P_{x_0 t}(X)$ lies in X_0. ∎

CLAIM 9. *At the time step in which* `hierarchical-petal-decomposition` *is called on* $(G(X), x_0, t)$, *the edges of* $G(X)$ *that are set to* $1/2$ *of their original weight are exactly those on* $P_{x_0 t}(X)$.

Proof. We will prove by induction on the depth of the recursion of `hierarchical-petal-decomposition`. The base case is trivial, since the first call to the algorithm is on (G, x_0, x_0). Assume by induction that X with center x_0 has a target t, and only edges on $P_{x_0t}(X)$ are set to 1/2 of their weight. We partition X into $X_0, X_1, \ldots, X_s$, and for $j \geq 1$ (or $j \geq 2$ if there is a special petal), set the weight of each edge on $P_{x_jt_j}(X)$ to 1/2 of its original weight. First, observe that these edges' weights haven't been changed before, because by the induction hypothesis on X the only edges set to 1/2 of their weight lie on $P_{x_0t}(X)$, but by Claim 8 all of them are fully contained in $G(X_0)$ or in $G(X_0 \cup X_1)$ if X_1 is a special petal.

Consider now X_0 with center x_0: if there is no special petal, then $t_0 = t \in X_0$, and by the induction hypothesis, the only edges in $G(X_0)$ that are set to 1/2 of their weight are those on $P_{x_0t_0}(X)$. If there is a special petal X_1, then $t_0 = y_1$, and as y_1 lies on P_{x_0t}, all the edges on $P_{x_0t_0}(X)$ are already set to 1/2 of their original weights.

If there is a special petal X_1, then its target is $t_1 = t$, and by the induction hypothesis, only the path $P_{x_1t_1} \subseteq P_{x_0t}(X)$ (which lies in $G(X_1)$ by Observation 3) contains edges that were already set to 1/2 of their original weight.

For all of the other petals X_j, we noted that none of the edges of $P_{x_0t}(X)$ (that were already set to 1/2 of their weight) are in X_j, and the assertion holds by construction. ☐

Corollary 10. *Every edge $e \in E$ can have its weight multiplied by 1/2 at most once throughout the running of the algorithm.*

Proof. Consider running `petal-decomposition`$(G(X), x_0, t)$. By Claim 8 the path $P_{x_0t}(X)$ is fully contained in $G(X_0)$ (or $G(X_0 \cup X_1)$ if X_1 is a special petal); thus it is disjoint from all the regular petals created. Observe that the algorithm changes the weights only for edges on $P_{x_jt_j}(X)$ for regular petals X_j, and by Observation 3, $P_{x_jt_j}(X) \subseteq G(X_j)$, so Claim 9 implies that these edges have not been reweighted before. ☐

Claim 11. *The algorithm returns a tree.*

Proof. Assume by induction on the depth of the recursion of `hierarchical-petal-decomposition` called on $G(X)$ that a tree is created. The base case when $|X| = 1$ trivially holds. Let $X \subseteq V$ be a cluster that is partitioned by the `petal-decomposition` algorithm into $X_0, X_1, \ldots, X_s$. By the induction hypothesis, running the algorithm on every subgraph $G(X_j)$ returns a tree T_j. Since every T_j contains $|X_j| - 1$ edges and we add s edges to create T, the total number of edges in the tree T created from X is indeed $|X| - 1$. It remains to show that there are no cycles. Seeking a contradiction, assume that there is a cycle. Since the edges $\{x_1, y_1\}, \ldots, \{x_s, y_s\}$ are not inside any cluster X_j, it must be that the cycle is not fully contained in a single X_j. Let $h \geq 1$ be the minimal integer such that the cycle contains vertices from X_h; thus there are at least two cycle edges leaving X_h. Observe that every edge $\{x_j, y_j\}$ we added satisfies $y_j \in Y_j$; in particular, if $j \geq h$, then $y_j \notin X_h$. This means that only $\{x_h, y_h\}$ can connect X_h to the other clusters in the cycle, which is a contradiction. ☐

4. Radius bound. Fix any cluster $X \subseteq V$ generated at some point during the execution of the algorithm. Let $T[X]$ be the tree created by calling `hierarchical-petal-decomposition`$(G(X), x_0, t)$. In this section we show that the radius of $T[X]$ is bounded by a constant times the radius of X. The metric d_X is the shortest path metric on X with respect to the *new* edge weights that were set just before calling `hierarchical-petal-decomposition`$(G(X), x_0, t)$. Recall that by Claim 9, the specific edges on $P_{x_0t}(X)$ had their weight set to 1/2 of the original weight. Also the metric d_T for any tree T is with respect to the new weights (note the tree is formed once the algorithm terminates). We start off by showing that this highway path is fully contained in $T[X]$.

Claim 12. *Let X be a cluster with center x_0 and target t; then*

$$d_T(x_0,t) = d_X(x_0,t)\ . \tag{12}$$

Proof. It suffices to show that the shortest path is $P_{x_0t}(X) \subseteq T$ and there is no reweighting on these edges. This can be shown by induction on the depth of the recursion of `hierarchical-petal-decomposition`. If there is no special petal created when calling `petal-decomposition` on X, then by Claims 1 and 8, $P_{x_0t}(X) = P_{x_0t}(X_0) \subseteq G(X_0)$, and by the induction hypothesis indeed $P_{x_0t}(X) \subseteq T$ without reweighting. Otherwise, if there is a special petal X_1, then $P_{x_0t}(X) \subseteq G(X_0 \cup X_1)$. Since we set $t_0 = y_1$ and $t_1 = t$, by the induction hypothesis both $P_{x_0y_1}(X_0) \subseteq T$ and $P_{x_1t}(X_1) \subseteq T$ without reweighting (there is no reweighting on a special petal). Also by definition the edge $\{y_1, x_1\}$ is added to T. Finally, recall that both x_1 and y_1 are chosen on $P_{x_0t}(X)$, so again by Claim 1 we obtain that $P_{x_0t}(X) = P_{x_0y_1}(X_0) \cup \{y_1, x_1\} \cup P_{x_1t}(X_1) \subseteq T$. □

Lemma 13. *For any cluster X created at some point during the algorithm,*

$$\mathrm{rad}(T[X]) \le 2\mathrm{rad}(X)\ .$$

Proof. It suffices to prove by induction on depth of the recursion of `hierarchical-petal-decomposition` that for any cluster X with center x_0 and target t, and for any $y \in X$,

$$d_T(x_0,y) \le 2d_X(x_0,y)\ . \tag{13}$$

Assume X is partitioned into clusters $X_0, X_1, \ldots, X_s$. There are three cases to consider: $y \in X_0$, $y \in X_1$ where X_1 is a special petal, and $y \in X_j$ for a regular petal X_j. Before showing the formal proof, the following is a high level description of these cases. Case 1 follows trivially by induction. Case 2 requires us to exploit the highway leading to the special petal, that is, the path from x_0 to the first petal will surely appear in the tree by Claim 12. The third case crucially uses the definition of petals and the reweighting of the highways. For every point y in a petal, the reweighting of the petal highway leading to y compensates for the increased distance incurred by its location in the petal.

Case 1. $y \in X_0$. By Claim 1, $P_{x_0y}(X) \subseteq X_0$. Applying the induction hypothesis on X_0 we obtain that $d_T(x_0,y) \le 2d_{X_0}(x_0,y) = 2d_X(x_0,y)$. The equality holds since the shortest path $P_{x_0y}(X_0)$ is the same as $P_{x_0y}(X)$ by Claim 1 (recall that there is no reweighting of edges in X_0). This concludes the first case.

In the other two cases, $y \in X_j$ for some $j \ge 1$. We now introduce some notation and show properties that hold in these two cases. Let r_j be the radius chosen by `create-petal` for creating X_j. Fix any $0 \le k \le r_j$ such that $y \in B_{(Y_{j-1},\rho(Y_{j-1},x_0,p_k))}(p_k,(r_j-k)/2)$ (note that k is not necessarily unique). By definition of a ball in a cone-metric (and using Claim 1 for distances involving x_0),

$$d_X(x_0,p_k) + d_{Y_{j-1}}(p_k,y) \le d_X(x_0,y) + (r_j-k)/2\ . \tag{14}$$

We shall use the following observations:

$$d_{X_j}(p_k,y) \le d_{Y_{j-1}}(p_k,y), \tag{15}$$

$$d_X(x_0,x_j) + d_X(x_j,p_k) = d_X(x_0,p_k)\ . \tag{16}$$

Recall that d_X is the metric induced on X with the *new* weights, occurring just before decomposing X. We denote $d_{Y_{j-1}}$ as the induced metric with weights as in X. To see (15), note that when taking a cone in the metric Y_{j-1} centered at p_k that contains y, it must also contain the entire shortest path from p_k to y, $P_{p_k y}(Y_{j-1})$. The inequality follows because distances in X_j can be made shorter due to reweighting of the edges on $P_{x_j t_j}$. For (16), this is simply because x_j is $p_{r'_j}$, all p_k are on the shortest path from t_j to x_0, and by definition of r'_j, $k \le r'_j$.

Case 2. There is a special petal X_1 and $y \in X_1$. In this case (15) implies that

$$d_{X_1}(x_1, p_k) \le d_X(x_1, p_k)\ . \tag{17}$$

By Claim 5 it follows that $d_X(x_0, x_1) \ge \Delta/2$, and thus

$$d_X(x_1, p_k) < \Delta/2\ . \tag{18}$$

Since a special petal was created we have that

$$r_1 \le d_X(x_0, t) - \Delta/2\ . \tag{19}$$

Recall that all the edges on $P_{x_0 t}(X)$ are already set to $1/2$ of their weight in X, and that y_1, x_1, p_k are all on $P_{x_0 t}$, so then $t_0 = y_1$, and by the induction hypothesis on X_1,

$$\begin{aligned}
d_T(x_0, y) &\le d_T(x_0, y_1) + d_T(y_1, x_1) + d_T(x_1, y) \\
&\overset{(12)\wedge(13)}{\le} d_X(x_0, y_1) + d_X(y_1, x_1) + 2d_{X_1}(x_1, y) \\
&\overset{(15)}{\le} d_X(x_0, x_1) + 2d_X(x_1, p_k) + 2d_X(p_k, y) \\
&\overset{(14)}{\le} d_X(x_0, p_k) + d_X(x_1, p_k) + 2(d_X(x_0, y) + (r_1 - k)/2 - d_X(x_0, p_k)) \\
&= 2d_X(x_0, y) + r_1 + d_X(x_1, p_k) - (d_X(x_0, p_k) + k) \\
&\overset{(18)\wedge(19)}{\le} 2d_X(x_0, y) + d_X(x_0, t) - \Delta/2 + \Delta/2 - d_X(x_0, t) \\
&= 2d_X(x_0, y)\ .
\end{aligned}$$

This concludes the proof for the second case.

Case 3. $y \in X_j$ for some (regular) petal X_j. Let us introduce some more notation. The *petal-tree* of a petal-decomposition on a subgraph $G(X)$ is a graph $H = (W, F)$, where $W = \{X_0, X_1, \ldots, X_s\}$ and $\{X_h, X_{h'}\} \in F$ iff $y_h \in X_{h'}$ or $y_{h'} \in X_h$ (that is, if the clusters are connected by one of the portal edges). Claim 11 suggests that H is a tree. Let X_0 be the root of the tree, and let $\text{rank}(X_h)$ denote the depth of X_h in H. Observe that in the case of regular petals (all petals other than the special one and the stigma), we have the following:

$$d_X(x_j, p_k) = 2d_{X_j}(x_j, p_k). \tag{20}$$

This holds because when we partition X, we assign all edges on $P_{x_j t_j}(X)$ half of their original weights; also note that Claim 9 suggests that the edges along $P_{x_j t_j}(X)$ were not reweighted before. By Observation 3 the shortest path $P_{x_j t_j}(X) \subseteq G(X_j)$, and the reweighting reduces the shortest path distance by a factor of 2.

We will prove (13) in the case $y \in X_j$, for some regular petal X_j, by induction on $\text{rank}(X_j)$. The base case is when the rank is 1 (rank 0 was handled in Case 1) so then it must be that $y_j \in X_0$. By Claim 1 we get $P_{x_0 y_j}(X) \subseteq G(X_0)$, and there is no reweighting of edges along $P_{x_0 y_j}$, so

$$d_{X_0}(x_0, y_j) = d_X(x_0, y_j) \ . \tag{21}$$

Note that the edge $\{y_j, x_j\}$ will never have its weight reduced by 1/2, because by Claim 9 this did not happen until now, and the edge will not be included in any future cluster. This means that

$$d_X(y_j, x_j) = d_T(y_j, x_j) \ . \tag{22}$$

By the induction hypothesis (13) on both X_0 and X_j,

$$\begin{aligned} d_T(x_0, y) &\leq d_T(x_0, y_j) + d_T(y_j, x_j) + d_T(x_j, y) \\ &\overset{(13)\wedge(22)}{\leq} 2d_{X_0}(x_0, y_j) + d_X(y_j, x_j) + 2d_{X_j}(x_j, y) \\ &\overset{(21)}{\leq} 2d_X(x_0, x_j) + 2d_{X_j}(x_j, p_k) + 2d_{X_j}(p_k, y) - d_X(y_j, x_j) \\ &\overset{(20)\wedge(15)}{\leq} 2d_X(x_0, x_j) + d_X(x_j, p_k) + 2d_{Y_{j-1}}(p_k, y) - d_X(y_j, x_j) \\ &= 2(d_X(x_0, p_k) + d_{Y_{j-1}}(p_k, y)) - d_X(y_j, p_k) \\ &\overset{(14)}{\leq} 2d_X(x_0, y) + (r_j - k) - (r_j - k) \\ &= 2d_X(x_0, y) \ . \end{aligned} \tag{23}$$

Now we prove for the case $\text{rank}(X_j) > 1$. Let $h \in [s]$ be such that $(X_j, X_h) \in F$ and $\text{rank}(X_h) = \text{rank}(X_j) - 1$. Observe that h is unique since H is a tree, and by definition of rank we have $y_j \in X_h$. By the induction hypothesis on the rank,

$$d_T(x_0, y_j) \leq 2d_X(x_0, y_j) \ . \tag{24}$$

And exactly the same calculation as in (23) holds, with the slight difference that the reason $d_T(x_0, y_j) \leq 2d_X(x_0, y_j)$ holds is by induction on the rank, rather than on X_0. This concludes the inductive proof. ∎

5. Analysis of total stretch. Recall that we apply `hierarchical-petal-decomposition` on the graph $G = (V, E)$ with center and target x_0. We wish to prove that the total stretch is bounded by $O(m \log n \log\log n)$. The proof is similar to the proof of [EEST05]. Consider a single run of the algorithm `create-petal` on input $(X, Y, t, x_0, [lo, hi])$. In what follows we shall omit the subscript Y. Recall that $R = hi - lo$, and let $q \geq 1$ be the value for which the loop terminated. We observe that $q \leq L$, because when $q = L$, the term in the right-hand side at line 4b of the algorithm is equal to $2m + 1$, but for any r, $\text{vol}(W_r) \leq 2m + 1$ (the last bound follows by (2)). Let $a = lo + (q-1)R/L$ and $b = lo + qR/L$. Recall that $\chi = \frac{|E(X)|+1}{\text{vol}(W_a)}$ and that r is chosen from $[a, b]$. The following claim is proved by a standard region growing argument (see, e.g., [Bar04]).

CLAIM 14. *The algorithm `create-petal` can find $r \in [a, b]$ satisfying*

$$\text{cost}(\partial(W_r)) \leq \text{vol}(W_r) \cdot (2L \ln \chi)/R \ . \tag{25}$$

Proof. First, note that increasing r by some small $\delta > 0$ will increase the radius of W_r by $\delta/2$, and thus for any edge $e \in \partial(W_r)$, its contribution to the volume of W_r will increase by $\delta \cdot \text{cost}(e)/2$ (this is by the definition of volume, as long as $e \in \partial(W_{r+\delta})$).

Consider all the vertices $x_1, \dots, x_t \in W_b \setminus W_a$, ordered according to the first time they enter W_r. In other words, take $a \le k_1 \le \cdots \le k_t \le b$ such that for all $i \in [t]$, $x_i \in W_{k_i}$, but $x_i \notin W_{k_i - \epsilon}$ for any $\epsilon > 0$. Define the functions $U : [a, b] \to \mathbb{R}$ and $c : [a, b] \to \mathbb{R}$ by $U(r) = \mathrm{vol}(W_r)$ and $c(r) = \mathrm{cost}(\partial(W_r))$. We recall that for all $r \in [a, b] \setminus \{k_1, \dots, k_t\}$, U is differentiable at r, and its derivative is $c(r)/2$. If we choose r uniformly at random from $[a, b]$, then putting $k_0 = a$ and $k_{t+1} = b$ we get that

$$\begin{aligned}
\mathbb{E}_r\left[\frac{c(r)}{U(r)}\right] &= \frac{1}{b-a}\sum_{i=0}^{t}\int_{k_i}^{k_{i+1}} \frac{c(r)}{U(r)}dr \\
&= \frac{2}{b-a}\sum_{i=0}^{t}\int_{k_i}^{k_{i+1}} \frac{U'(r)}{U(r)}dr \\
&= \frac{2}{b-a}\sum_{i=0}^{t}[\ln U(r)]_{k_i}^{k_{i+1}} \\
&= \frac{2}{b-a}\sum_{i=0}^{t}(\ln U(k_{i+1}) - \ln U(k_i)) \\
&= \frac{2}{b-a}(\ln U(b) - \ln U(a)) \\
&\le (2L \ln \chi)/R \ .
\end{aligned}$$

The last inequality uses that $\mathrm{vol}(W_b) \le |E(X)| + 1$. We conclude that there exists $r \in [a, b]$ such that

$$\mathrm{cost}(\partial(W_r)) \le \mathrm{vol}(W_r) \cdot (2L \ln \chi)/R,$$

and since $c(r)$ remains constant in each interval (k_i, k_{i+1}), such an r can be found efficiently by checking the values at $r = k_i$ for $i \in [t]$. ◻

Consider now the algorithm `petal-decomposition` invoked on some $G(X)$ with center x_0 and target t. It decomposes X into $X_0, X_1, \dots, X_s$ (for some integer $s \ge 1$). For $j \in [s]$, let χ_j be the value defined at line 4a of `create-petal` when creating the petal X_j, and denote by $\mathrm{index}(X_j)$ the value of q for which the loop is terminated. For $q > 1$, since the loop did not stop at $q - 1$ we have that $\mathrm{vol}(W_a) > \frac{2|E(X)|}{2^{\log^{1-(q-1)/L} m}} + 1$; in particular, as long as $|E(X)| \neq 0$, we have $\chi_j = \frac{|E(X)|+1}{\mathrm{vol}(W_a)} \le 2^{\log^{1-(q-1)/L} m}$. Thus we have

$$\ln \chi_j \le \log^{1-(q-1)/L} m + 1 \le 3 \log^{1-q/L} m \tag{26}$$

(the last inequality is because $\log^{1/L} m = 2^{\log\log m/\log\log n} \le 5/2$). Observe that (26) holds also for $q = 1$, because $\mathrm{vol}(W_a) \ge 1$ and $|E(X)| \le m$. Also, if the degenerate case $|E(X)| = 0$, we have $\ln \chi_j = 0$, so the inequality holds. Note that if some edge $\{u, v\} \in E(X)$ is separated while decomposing the cluster X with radius Δ (that is, it belongs to some $\partial(X_j)$), then by Lemma 13

$$d_T(u, v) \le 2\mathrm{rad}(T[X]) \le 4\Delta \ ,$$

and by Corollary 10 the distance in T with respect to the *original* weights is at most twice as large. We conclude that

$$\mathrm{stretch}_T(u, v) \le 8\Delta/w(u, v) = 8\mathrm{cost}(u, v) \cdot \Delta \ . \tag{27}$$

We now start to calculate the total stretch. Every separated edge appears in some $\partial(X_j)$ for $j \geq 1$; recalling that $R = \Delta/8$ we get

$$\begin{aligned}
\mathrm{TS}[X] &\overset{(27)}{\leq} \mathrm{TS}[X_0] + \sum_{j=1}^{s} (\mathrm{TS}[X_j] + \mathrm{cost}(\partial(X_j)) \cdot 8\Delta) \\
&\overset{(25)}{\leq} \mathrm{TS}[X_0] + \sum_{j=1}^{s} \big(\mathrm{TS}[X_j] + 2^7 L \cdot \mathrm{vol}(X_j) \cdot \ln \chi_j\big) \\
&\overset{(26)}{\leq} \mathrm{TS}[X_0] + \sum_{j=1}^{s} \mathrm{TS}[X_j] + 2^{10} L \cdot \sum_{q=1}^{L} \sum_{j:\mathrm{index}(X_j)=q} \mathrm{vol}(X_j) \cdot \log^{1-q/L} m. \qquad (28)
\end{aligned}$$

Let us fix some edge $e \in E$ and analyze its contribution to (28). For every level of the recursion in which e participated in $\mathrm{vol}(X_j)$ with $q = \mathrm{index}(X_j)$ it contributes $O(L) \cdot \log^{1-q/L} m$. However, by the choice of q, $|E(X_j)| \leq \mathrm{vol}(X_j) \leq \frac{4|E(X)|}{2^{\log^{1-q/L} m}} + 1$. Intuitively, if q is small and thus the contribution is rather large, the volume of the next cluster that contains e becomes much smaller, so e will participate in only a few more levels. In particular, if the contribution to the total stretch of e in some level is $O(L \cdot i)$, then the number of edges in the cluster containing e is reduced by a factor of $\Omega(2^i)$. Since the number of times the number of edges can halve is at most $O(\log m)$, we get that the total contribution of each edge over all levels is at most $O(L \cdot \log m) = O(\log n \log \log n)$. We now show this formally. Let $\ell_q(e)$ denote the number of recursive levels i in which e was contained in a cluster of index q. Then the number of edges in the clusters containing e decreased by a factor of at least $2^{\log^{1-q/L} m - 3}$ for every one of the $\ell_q(e)$ levels (assuming $4|E(X)| \geq 2^{\log^{1-q/L} m}$, we indeed have that $|E(X_j)| \leq \frac{8|E(X)|}{2^{\log^{1-q/L} m}}$, but otherwise the cluster X_j contains at most 1 edge and the recursion will terminate the next level). So the total decrease is

$$\prod_{q=1}^{L} 2^{\ell_q(e) \cdot (\log^{1-q/L} m - 3)} \leq m \ ,$$

because we started with m edges. This suggests that

$$\sum_{q=1}^{L} \ell_q(e)(\log^{1-q/L} m) \leq \log m + 3 \sum_{q=1}^{L} \ell_q(e) = O(\log m) \ ,$$

where we used that e participates in $O(\log n)$ levels, that is, $\sum_{q=1}^{L} \ell_q(e) = O(\log m)$ (this follows by the edge contractions; see below in the beginning of section 6). Also note that when e is cut it, in fact, contributes twice to (28), but since this can happen only once throughout the recursion, and the contribution is at most $O(L \cdot \log m)$ for each edge, this adds an additional $O(Lm \log n)$ factor to the total stretch. Finally,

$$\begin{aligned}
\mathrm{TS}[V] \leq O(L) \sum_{e \in E} \sum_{q=1}^{L} \ell_q(e) \log^{1-q/L} m + O(Lm \log n) &= O(Lm \log m) \\
&= O(m \log n \log \log n) \ .
\end{aligned}$$

6. Fast petal construction.

Handling arbitrary weights. We first give some details on handling graphs with arbitrary weights (so far we assumed the aspect ratio is polynomial in n). We use a well-known trick (see, e.g., [EEST05]) when handling a cluster X, contract all the edges in $G(X)$ of weight smaller than $\mathrm{rad}(X)/n^2$. As at most $n-1$ edges are contracted, distances in X and in its children $X_0, \ldots, X_s$ can differ by at most $\mathrm{rad}(X)/n$. So we ignore this small factor when bounding the radius in section 4.

Running time. In order to bound the running time of our algorithm, we need to argue that the petal construction can be performed efficiently. We first show that by the contraction and expansion of edges, each edge participates in a logarithmic number of levels. Consider any edge $e \in E$. Since e cannot be contained in clusters of radius less than $w(e)/2$ and is contracted in clusters of radius greater than $n^2 \cdot w(e)$, we have that e participates only at partitions of radius in the range $[w(e)/2, n^2 \cdot w(e)]$. By Claims 5 and 6 the radii decrease by a factor of at least $3/4$ at every level, so there are at most $O(\log n)$ levels in which each edge participates.

It is shown in [KMP11] how to construct a star-decomposition on $G = (V, E)$ in time $O(|E| + |V| \log k)$, where k is the number of distinct edge weights. This factor essentially comes from running an improved version of Dijkstra's algorithm for computing the shortest path from the center of the cluster, as introduced by [OMSW10]. By rounding down weights to the nearest power of 2, we change distances by a factor of 2, and in every level there will be at most $O(\log n)$ different edge weights. As there are $O(\log n)$ scales in which any edge is active, we conclude that the total running time will be $O((m + n \log\log n) \cdot \log n)$. It remains to determine whether petals, much like cones, can be constructed efficiently as a region growing scheme.

Given a weighted undirected graph $G = (V, E)$ with a center x and a target t, let p_k be the point of distance k from t on the shortest path P_{tx} from t to x; all distances d are with respect to G. Let $\tilde{G} = (V, A, \bar{w})$ be the weighted directed graph induced by adding the two directed edges $(u, v), (v, u) \in A$ for each $\{u, v\} \in E$ and setting $\bar{w}(u, v) = w(u, v) - (d(v, x) - d(u, x))$. (Note that $d(v, x) - d(u, x) \leq d(u, v) \leq w(u, v)$, so the weights are positive.) We can efficiently compute all of these weights by running the improved Dijkstra from the center x in G. The cone $C(x, t, r)$ is simply the ball around t of radius r in $\tilde{G}$. The Petal $P(x, t, r)$ is the ball around t of radius $r/2$ in $\tilde{G}$ with one change: the weight of each edge on the path $P_{t,p_{r'}}$ is changed to be $1/2$ of its original weight (that is, $1/2$ of its weight in G), where $r' \leq r$ is the maximal such that there exists a point $p_{r'}$ on this path. Recall that the petal with center x, target t, and radius r were defined in the algorithm as $W_r = \bigcup_{p \in P_{xt} \,:\, d(p,t) \leq r} B_{(V, \rho(V,x,p))}(p, (r - d(p, t))/2)$. It remains to see that both definitions yield the same result, and thus to compute the petals we can use the improved Dijkstra on the graph $\tilde{G}$.

Claim 15. $P(x, t, r) = W_r$.

Proof. First, we prove that for any $r \geq 0$, $W_r \subseteq P(x, t, r)$. Fix some $v \in W_r$, and let $0 \leq k \leq r$ be such that $v \in B_{(V, \rho(V,x,p_k))}(p_k, (r - k)/2)$. Observe that by the reweighting of the edges from t to p_k we have that the length of the directed path P_{tp_k} in $\tilde{G}$ is $k/2$. It remains to show that there is a path in $\tilde{G}$ from p_k to v of length at most $(r - k)/2$. By definition of cone-metric we have that $d(v, p_k) + d(p_k, x) \leq d(v, x) + (r - k)/2$. Let $p_k = u_0, u_1, \ldots, u_l = v$ be the shortest path in G from p_k to

v; then by definition of $\bar{w}$ it follows that

$$\sum_{j=1}^{l} \bar{w}(u_{j-1}, u_j) = \sum_{j=1}^{l} w(u_{j-1}, u_j) - d(u_j, x) + d(u_{j-1}, x)$$
$$= d(p_k, v) - d(v, x) + d(p_k, x) \leq (r-k)/2 \ ,$$

as required.

Next, we prove that $P(x,t,r) \subseteq W_r$ by induction on the radius. Let $0 = k_1 < k_2 < \cdots < k_l$ be all the possible values of r for which the size of $P(x,t,r)$ changes. The base case for $k_1 = 0$, then $W_0 = \{y \ : \ d(y,x) = d(y,t) + d(t,x)\}$, and then $P(x,t,0)$ will contain all points reachable with 0 weight edges; by definition these edges (u,v) are the ones that satisfy $d(v,x) - d(u,x) = w(u,v)$, so any path leaving t using these edges will lead to a point y for which $d(y,x) = d(y,t) + d(t,x)$, and so $y \in W_0$.

For the inductive step, assume $P(x,t,k_{i-1}) \subseteq W_{k_{i-1}}$ and prove for k_i. Let $\delta = k_i - k_{i-1}$. Let $v \in P(x,t,k_i) \setminus P(x,t,k_{i-1})$, and assume $u \in P(x,t,k_{i-1})$ is such that $(u,v) \in A$ with $\bar{w}(u,v) \leq \delta/2$. Then by definition of $\bar{w}$ we have that $d(u,v) \leq w(u,v) \leq \delta/2 + d(v,x) - d(u,x)$. By the induction hypothesis we have that $u \in W_{k_{i-1}}$, so let k be such that $u \in B_{(V,\rho(V,x,p_k))}(p_k, (k_{i-1}-k)/2)$. By definition of cone-metric $d(u,p_k) + d(p_k,x) \leq d(u,x) + (k_{i-1}-k)/2$. It follows that

$$d(v,p_k) + d(p_k,x) \leq d(v,u) + d(u,p_k) + d(p_k,x)$$
$$\leq \delta/2 + d(v,x) - d(u,x) + d(u,x) + (k_{i-1}-k)/2$$
$$= d(v,x) + (k_i - k)/2 \ ,$$

meaning that $v \in W_{k_i}$. ∎

Acknowledgments. We would like to thank Yair Bartal, Michael Elkin, and Kunal Talwar for helpful discussions, and we thank Arnold Filtser and Benny Kramer for a careful reading of this paper and making useful suggestions.

REFERENCES

[ABN06] I. Abraham, Y. Bartal, and O. Neiman, *Advances in metric embedding theory*, in Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing, STOC '06, ACM, New York, 2006, pp. 271–286.

[ABN07] I. Abraham, Y. Bartal, and O. Neiman, *Embedding metrics into ultrametrics and graphs into spanning trees with constant average distortion*, in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, SIAM, Philadelphia, 2007, pp. 502–511.

[ABN08] I. Abraham, Y. Bartal, and O. Neiman, *Nearly tight low stretch spanning trees*, in FOCS '08: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Soc., Washington, DC, 2008, pp. 781–790.

[ACE+18] I. Abraham, S. Chechik, M. Elkin, A. Filtser, and O. Neiman, *Ramsey spanning trees and their applications*, in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, SIAM, Philadelphia, 2018, pp. 1650–1664.

[AF09] R. Andersen and U. Feige, *Interchanging Distance and Capacity in Probabilistic Mappings*, https://arxiv.org/abs/0907.3631, 2009.

[AKPW95] N. Alon, R. M. Karp, D. Peleg, and D. West, *A graph-theoretic game and its application to the k-server problem*, SIAM J. Comput., 24 (1995), pp. 78–100, https://doi.org/10.1137/S0097539792224474.

[Bar96] Y. Bartal, *Probabilistic approximation of metric spaces and its algorithmic applications*, in Proceedings of the 37th Annual Symposium on Foundations of Computer Science, IEEE Computer Soc., Washington, DC, 1996, pp. 184–193.

[Bar98] Y. Bartal, *On approximating arbitrary metrices by tree metrics*, in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC '98, ACM, New York, 1998, pp. 161–168.

[Bar04] Y. Bartal, *Graph decomposition lemmas and their role in metric embedding methods*, in Algorithms—ESA, S. Albers and T. Radzik, eds., Lecture Notes in Comput. Sci. 3221, Springer, Berlin, 2004, pp. 89–97.

[BIS07] M. Bădoiu, P. Indyk, and A. Sidiropoulos, *Approximation algorithms for embedding general metrics into trees*, in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, SIAM, Philadelphia, 2007, pp. 512–521.

[CCGG98] M. Charikar, C. Chekuri, A. Goel, and S. Guha, *Rounding via trees: Deterministic approximation algorithms for group steiner trees and k-median*, in STOC '98: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 114–123.

[CDN+10] V. Chepoi, F. F. Dragan, I. Newman, Y. Rabinovich, and Y. Vaxès, *Constant approximation algorithms for embedding graph metrics into trees and outerplanar graphs*, in Approximation, Randomization, and Combinatorial Optimization, Lecture Notes in Comput. Sci. 6302, Springer, Berlin, 2010, pp. 95–109.

[CKM+14] M. B. Cohen, R. Kyng, G. L. Miller, J. W. Pachocki, R. Peng, A. B. Rao, and S. C. Xu, *Solving sdd linear systems in nearly* $mlog1/2n$ *time*, in Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC '14, ACM, New York, 2014, pp. 343–352.

[EEST05] M. Elkin, Y. Emek, D. A. Spielman, and S.-H. Teng, *Lower-stretch spanning trees*, in STOC '05: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, ACM Press, New York, 2005, pp. 494–503.

[EFN15] M. Elkin, A. Filtser, and O. Neiman, *Terminal embeddings*, in Approximation, Randomization, and Combinatorial Optimization, LIPIcs. Leibniz Int. Proc. Inform, 40, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, Germany, 2015, pp. 242–264.

[EP04] Y. Emek and D. Peleg, *Approximating minimum max-stretch spanning trees on unweighted graphs*, in Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '04, SIAM, Philadelphia, 2004, pp. 261–270.

[EP06] Y. Emek and D. Peleg, *A tight upper bound on the probabilistic embedding of series-parallel graphs*, in Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, SODA '06, ACM, New York, 2006, pp. 1045–1053.

[FRT03] J. Fakcharoenphol, S. Rao, and K. Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, in Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, STOC '03, ACM, New York, 2003, pp. 448–455.

[Hu74] T. C. Hu, *Optimum communication spanning trees*, SIAM J. Comput., 3 (1974), pp. 188–195, https://doi.org/10.1137/0203015.

[KLP+16] R. Kyng, Y. T. Lee, R. Peng, S. Sachdeva, and D. A. Spielman, *Sparsified cholesky and multigrid solvers for connection Laplacians*, in Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, ACM, New York, 2016, pp. 842–850.

[KMP11] I. Koutis, G. L. Miller, and R. Peng, *A nearly-m log n time solver for sdd linear systems*, in Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS '11, IEEE Computer Soc., Los Alamitos, CA, 2011, pp. 590–598.

[KOSZ13] J. A. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu, *A simple, combinatorial algorithm for solving sdd systems in nearly-linear time*, in Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, STOC '13, ACM, New York, 2013, pp. 911–920.

[KS16] R. Kyng and S. Sachdeva, *Approximate Gaussian elimination for Laplacians—fast, sparse, and simple*, in Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2016, IEEE Computer Soc., Los Alamitos, CA, 2016, pp. 573–582.

[McK15] R. L. McKnight, *Low-Stretch Trees for Network Visualization*, Ph.D. thesis, University of British Columbia, Vancouver, BC, 2015.

[OMSW10] J. B. Orlin, K. Madduri, K. Subramani, and M. Williamson, *A faster algorithm for the single source shortest path problem with few distinct positive lengths*, J. Discrete Algorithms, 8 (2010), pp. 189–198.

[Sey95] P. D. Seymour, *Packing directed circuits fractionally*, Combinatorica, 15 (1995), pp. 281–288.

[ST04] D. A. Spielman and S.-H. Teng, *Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems*, in Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, STOC '04, ACM, New York, 2004, pp. 81–90.