Using Petal-Decompositions to Build a Low Stretch Spanning Tree

Ittai Abraham[*] Ofer Neiman[†]

March 22, 2012

Abstract

We prove that any graph $G = (V, E)$ with n points and m edges has a spanning tree T such that $\sum_{(u,v)\in E(G)} d_T(u, v) = O(m \log n \log\log n)$. Moreover such a tree can be found in time $O(m \log n \log\log n)$. Our result is obtained using a new *petal-decomposition* approach which guarantees that the radius of each cluster in the tree is at most 4 times the radius of the induced subgraph of the cluster in the original graph.

[*]Microsoft Research SVC. Email: ittaia@microsoft.com.
[†]Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel. Email: neimano@cs.bgu.ac.il.

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

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

Let $\text{avg stretch}(n) = \max_{G=(V,E,w)\,:\,|V|=n} \inf_T\{\text{avg stretch}_T(G)\}$.

Figure 1 summarizes current progress on the bounds for $\text{avg stretch}(n)$ and the time complexity of building such trees.

	$\text{avg stretch}(n)$	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]	$O(\log n \log\log n(\log\log\log n)^3)$	$O(m^2)$
[KMPb]	$O(\log n(\log\log n)^3)$	$O(m \log n \log\log n)$
This paper	$O(\log n \log\log n)$	$O(m \log n \log\log n)$

Figure 1: Summary of Progress on Low Stretch Spanning Tree.

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

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

Theorem 1.

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

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

Our result may be applied to improve the running time of the Spielman and Teng [ST04] approach to solve sparse symmetric diagonally dominant linear systems, using the improved algorithms of Koutis, Miller and Peng [KMPa, KMPb].

1.1 Related Work

Embedding metric spaces and graphs into tree metrics and spanning trees has received a lot of attention in the last two decades. The basic motivation is that problems on simple graphs such as trees are often much easier than 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 a $\exp(O(\sqrt{\log n \log\log n}))$ bound on the average stretch. If we drop the requirement that the tree is spanning (that is, allow to add and not only delete edges, while maintaining that distances in the tree are larger 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 non-spanning 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, 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 ([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], in which 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*, that allows 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 into a tree, using some of the inter cluster edges, and all other edges between clusters are dropped.

The petal-decomposition approach differs from star-decompositions in three main aspects. First, it is not the case that all petals are necessarily connected to the stigma (as would be the case in the star-partition); petals are connected to each other in a tree structure whose root is the stigma. Second, the stigma is not necessarily a ball, it is the remaining subgraph once no more petals can be formed. Third and most important, is the definition of a petal. In a star-decomposition each cone $C(x_0, x, r)$ is defined by three parameters: 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 $d(x_0, x) + d(x, v) - d(x_0, v) \leq r$. The radius r of the cone determines the maximum increase in the radius of the graph. A petal $P(x_0, t, r)$ is also defined by three parameters:

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

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. The 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 $k \leq r$.

Informally, the crucial property of a petal and its highway is the following: Assume $z \in P(x_0, t, r)$, and $P_{x_0 z}$ 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 2. Such a property could allow the following wishful thinking: Suppose that in each iteration we increase the distance of a point to the center by at most α, but also mark 2α of the path 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 of at most $O(1)$ (intuitively, the highway part will quickly catch-up and the process stops when all the path is marked as highway). Unfortunately, the shortest path from x to z in the final tree may not use the prescribed highway of the parent cluster 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 re-weighting signals later iterations to either use the prescribed highway or to find an alternative path whose short length can compensate for the lack of using the prescribed 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 re-weighting 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 re-weighted 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. Note that one must always lose a factor of at least 2 for any spanning tree.

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. We shall elaborate more on this in Section 7.

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

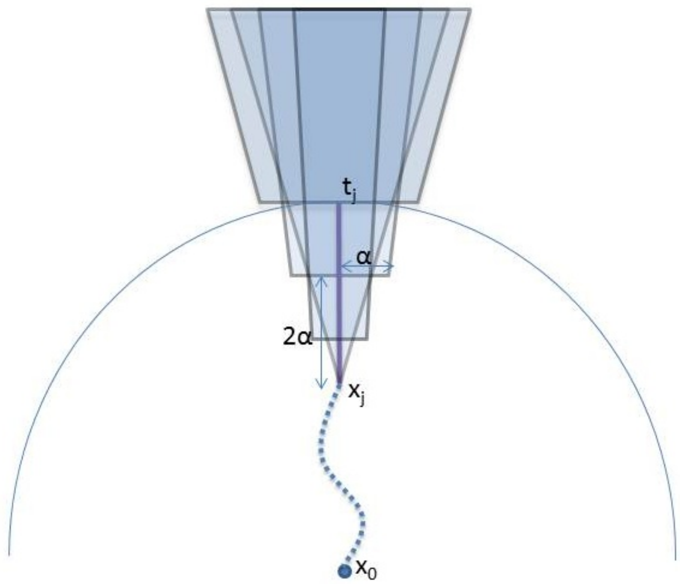

Figure 2: Definition of a petal with target t_j, center x_j and highway path $P_{x_j t_j}$. The side radius of each cone (that determines the maximum increase in the radius) is half of the highway path.

factor proportional to the diameter of the created tree. The construction has to balance between 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. [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. [ABN08] have 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]) and this would lead to optimal average stretch. Known strong diameter partitions ([ABN08]) that obtain a local growth ratio bound require to carefully choose the center of each cluster. However, our current petal decomposition approach does not allow to choose the centers arbitrarily and hence we could not use directly 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 for obtaining an optimal low stretch tree.

1.3 Applications

One of the most important problems in algorithm design is obtaining fast algorithms for solving linear systems. For many applications the matrix is sparse, and while little is known for general sparse matrices, the case of Symmetric Diagonally Dominant (SDD) matrices has received a lot of attention recently. In a seminal sequence of results, Spielman and Teng [ST04] showed a near linear time solver for this important case. This solver has proven a powerful algorithmic tool, and is used to calculate eigenvalues, obtain spectral

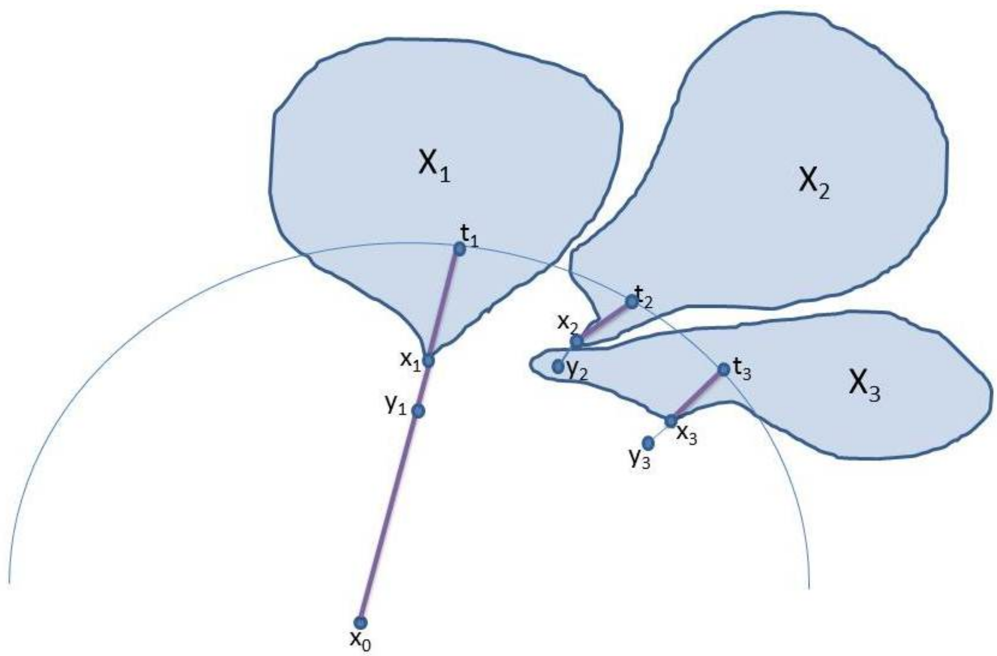

Figure 3: Creating the first three petals with their highways. The first portal is connected by a highway to x_0 (this means that the shortest path from x_0 to x_1 will be included in the final tree). Note that the portal edges do not necessarily connect the petal to the stigma, but may connect between 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).

graph sparsifiers [SS08], approximate maximum flow [CKM+11] and in many more applications. A basic step in solving these systems $Ax = b$ is combinatorial preconditioning. If one uses the Laplacian matrix corresponding to a spanning tree (and few extra edges) of the graph whose Laplacian matrix is A, then the condition number depends on the total stretch of the tree. This will improve the run-time of iterative methods, such as Conjugate Gradient or Chebyshev iterations. See [KMPa, KMPb] for the latest progress on this direction. In this work we show that one can construct such a spanning tree with both run-time and total stretch bounded by $O(m \log n \log\log n)$.

There are more applications for low stretch spanning trees, such as minimum cost communication spanning tree, we refer the reader to [EEST05, ABN08] for more details.

1.4 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 7 we show an alternative view of forming a petal, similar to region growing techniques that concludes the proof of Theorem 1. In Section 6 we discuss briefly how to extend the result to weighted graphs.

2 Preliminaries

Let $G = (V, E)$ be an unweighted undirected graph. For any $X \subseteq V$, $G(X)$ is the subgraph induced on X with edges $E(X) = \{(u, v) \in E \mid u, v \in X\}$. Denote by $E^+(X) = \{(u, v) \in E \ : \ |X \cap \{u, v\}| \geq 1\}$ the set of edges with at least one edge point in X, and by $\partial(X) = \{(u, v) \in E \ : \ |X \cap \{u, v\}| = 1\}$ the set of edges with exactly one end point in 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 clear from 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 $B_{(X,d_X)}(x, r) = \{y \in X \mid d_X(x, y) \leq r\}$.

For a spanning tree $T = T[X]$ of a subgraph X define the total stretch of T by

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

For $X \subseteq V$ and vertices $u, v \in X$, let $P_{uv} = P_{uv}(G(X))$ be a fixed shortest path between u, v in X (assuming that $G(X)$ is connected). We shall assume that there is a unique such path; This can be achieved, for example, by adding an imaginary random tiny amount to every edge length. Adding a path of length k starting at vertex v means that we set $v = u_0$, add new vertices $u_1, \ldots, u_k$ and add edges (u_{i-1}, u_i) for all $i \in \{1, \ldots, k\}$. By $T = \mathrm{BFS}_x(G(X))$ we mean the Breadth First Search tree rooted at x (the subscript is dropped when the center x is clear from context). Observe that for such a tree $d_T(x, y) = d_X(x, y)$ for all $y \in X$.

Definition 1 (cone metric[2])**.** *Given a graph $G = (V, E)$, subset $X \subseteq V$ and 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] (this one is less general). 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 $d_X(x, y) + d_X(y, z) - d_X(x, z) \leq r$.

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

$T = \texttt{hierarchical-petal-decomposition}(G(X), x_0, t)$:

1. If $\mathrm{rad}_{x_0}(X) \le 10 \log n \log\log n$ return BFS$(G(X))$.
2. $(X_0, \dots, X_s, (y_1, x_1), \dots, (y_s, x_s), t_0, \dots, t_s) = \texttt{petal-decomposition}(G(X), x_0, t)$;
3. For each $j \in [0, \dots, s]$:
   (a) Set all the edges in $P_{x_j t_j}$ to be of weight $1/2$;
   (b) $T_j = \texttt{hierarchical-petal-decomposition}(G(X_j), x_j, t_j)$;
4. Let T be the tree formed by connecting $T_0, \dots, T_s$ using the edges $(y_1, x_1), \dots, (y_s, x_s)$;

Figure 4: `hierarchical-petal-decomposition` algorithm

3 Petal-decomposition

Hierarchical-petal-decomposition algorithm. See Figure 4 for the algorithm. Let $G = (V, E)$ be an unweighted graph $G = (V, E)$. Here and in all that follows $n = |V|$ and $m = |E|$. Create a spanning tree $T = (V, E')$ by choosing some $x_0 \in V$ and calling $\texttt{hierarchical-petal-decomposition}(G, x_0, x_0)$.

3.1 Properties and Correctness

Fix some subset $X \subseteq V$, and consider running the `hierarchical-petal-decomposition` algorithm on $G(X)$ with some $x_0 \in X$ and target $t \in X$. Denote by $\Delta = \mathrm{rad}_{x_0}(X)$. Let r_j be the radius chosen by the algorithm `create-petal` when it is invoked to create petal X_j. In order to show that the algorithm is correct, we need to show the following: that a tree is created, that every cluster is connected, and that for all integers $1 \le j \le s$, $x_j, t_j \in X_j$. 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 give it for 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 $u \in X_h$. 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 $P_{x_h t_h}$ of distance k from t_h such that $u \in B_{Y_{h-1}, \rho(Y_{h-1}, x_0, p_k)}(p_k, (r_h - k)/2)$. By Definition 1 this means that

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

We claim that $P_{x_0 z}$ is fully contained in $G(Y_{h-1})$: if $h > 1$ then it holds by the minimality of h, otherwise, this holds as $Y_0 = X$. Since u lies on $P_{x_0 z}$, it follows that $d_{Y_{h-1}}(x_0, z) = d_{Y_{h-1}}(x_0, u) + d_{Y_{h-1}}(u, z)$. Now

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

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

1. Let $\Delta = \mathrm{rad}_{x_0}(X)$; Let $r_0 = \Delta/2$; $Y_0 = X$; Set $j = 2$;
2. Creating the first petal X_1:
   (a) If $d_X(x_0, t) < r_0$, add to $G(X)$ a path $(u_0, u_1, \ldots, u_l)$ of length $l = r_0 - d_X(x_0, t)$ starting at $t = u_0$; Let $t'_1 = t_1 = u_l$; Otherwise, let $t_1 = t$ and let t'_1 be a vertex on $P_{x_0 t}$ such that $d_X(x_0, t'_1) = r_0$;
   (b) Let $(X_1, x_1) = \texttt{create-petal}(X, X, t'_1, x_0, \Delta/4)$; $Y_1 = Y_0 \setminus X_1$;
   (c) Let y_1 be the neighbor of x_1 on $P_{x_0 t'_1}$ that is closer to x_0;
3. Creating the remaining petals $X_2, \ldots, X_s$:
   (a) While $Y_{j-1} \setminus B_X(x_0, r_0) \neq \emptyset$:
      i. Let $t_j \in Y_{j-1}$ be an arbitrary point satisfying $d_X(x_0, t_j) = r_0$;
      ii. Let $(X_j, x_j) = \texttt{create-petal}(X, Y_{j-1}, t_j, x_0, \Delta/8)$; $Y_j = Y_{j-1} \setminus X_j$;
      iii. Let y_j be the neighbor of x_j on $P_{x_0 t_j}$ that is closer to x_0;
      iv. Let $j = j + 1$;
   (b) Let $s = j - 1$;
4. Creating the stigma X_0:
   (a) Let $X_0 = Y_s$; Let $t_0 = y_1$;

Figure 5: `petal-decomposition` algorithm

$(W, x) = \texttt{create-petal}(X, Y, t, x_0, R)$:

1. Let $W_r = \bigcup_{p \in P_{x_0 t} \,:\, d_Y(p,t) \le r} B_{(Y, \rho(Y, x_0, p))}(p, (r - d_Y(p, t))/2)$;
2. Let $L = \lceil \log\log n \rceil$; Let $1 \le p \le L$ be the minimal integer satisfying $|E(W_{(1+p/L)R/2})| \le \frac{2|E(X)|}{2^{\log^{1-p/L} m}}$; Set $a = (1 + (p-1)/L) \cdot R/2$;
3. Set $r = a$; Fix $\chi = \frac{|E(X)|}{|E(W_a)|}$;
4. Increase r as long as $|\partial(W_r)| \ge |E(W_r)| \cdot \frac{8L \ln \chi}{R}$.
5. Return (W_r, p_r).

Figure 6: `create-petal` algorithm

hence $z \in B_{Y_{h-1},\rho(Y_{h-1},x_0,p_k)}(p_k,(r_h-k)/2)$ and thus also in X_h, 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 $j \geq 1$, $P_{x_j t_j}(X) \subseteq G(X_j)$.*

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 $p \in P_{x_j t_j}$. This suggests that the entire path $P_{x_j t_j}(X) \subseteq G(X_j)$. □

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

Proof. By Observation 3, $P_{x_j t_j}$ 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 $P_{x_j t_j}$, it is connected. □

Observation 5. *Algorithm* `create-petal`(X, Y, t, x_0, R) *chooses a radius $r \in [R/2, R]$.*

Proof. This follows from Claim 14. □

The following two claims are similar to claims proven in [EEST05, ABN08], we provide proofs for completeness.

Claim 6. $\Delta/4 \leq \mathrm{rad}_{x_0}(X_0) \leq \Delta/2$.

Proof. For the upper bound, note that for any $j \geq 1$, if there is a point in $Y_{j-1} \setminus B(x_0, \Delta/2)$ we continue creating petals, therefore $Y_s \setminus B(x_0,\Delta/2) = \emptyset$ and $X_0 = Y_s \subseteq B(x_0,\Delta/2)$.

To see the lower bound, observe that Claim 1 implies that for any $j \geq 1$, $d_{Y_{j-1}}(x_0,t_j) = r_0 = \Delta/2$, and also by Observation 5 we have that the radius r_j chosen for each of the petals, satisfies $r_j \leq \Delta/4$. Consider some $z \in X$ with $d_X(x_0,z) < \Delta/4$, we will show that $z \in X_0$. For any $0 \leq k \leq r_j$ and $p_k \in P_{x_j t_j}$ of distance k from t_j,

$$\begin{aligned}
d_{Y_{j-1}}(z,p_k) + d_{Y_{j-1}}(p_k,x_0) &= d_{Y_{j-1}}(z,p_k) + d_X(p_k,x_0) \\
&\geq d_X(p_k,x_0) \\
&\geq \Delta/2 - k \\
&\geq \Delta/4 + (\Delta/4 - k)/2 \\
&> d_X(x_0,z) + (r_j - k)/2\ .
\end{aligned}$$

By the definition of cone metric, this implies that $z \notin X_j$, for all $j \geq 1$. □

Claim 7. *For each $1 \leq j \leq s$, $\mathrm{rad}_{x_j}(X_j) \leq 7\Delta/8$.*

Proof. Fix some integer $1 \leq j \leq s$. We already know by Observation 3 that $P_{x_j t_j}(X) \subseteq G(X_j)$, recall that the petal X_j is created by union over balls (in a cone-metric) centered at the points of $P_{x_j t_j}(X)$. By Observation 5 the radius of each ball is bounded by $r_j \leq \Delta/4$, and we also have by Claim 6 that

$$d_X(x_0,x_j) \geq \Delta/4\ . \tag{3}$$

Let $y \in X_j$, we will show that $d_{X_j}(x_j, y) \le 7\Delta/8$. Let $0 \le k \le r_j$ and $p_k \in P_{x_j t_j}$ of distance k from t_j such that $y \in B_{(Y_{j-1}, \rho(Y_{j-1}, x_0, p_k))}(p_k, (r_j - k)/2)$. By definition of cone-metric and using Claim 1

$$\begin{aligned} d_X(x_0, p_k) + d_{Y_{j-1}}(p_k, y) &= d_{Y_{j-1}}(x_0, p_k) + d_{Y_{j-1}}(p_k, y) &(4)\\ &\le d_{Y_{j-1}}(x_0, y) + (r_j - k)/2 &(5)\\ &= d_X(x_0, y) + (r_j - k)/2 &(6)\\ &\le 9\Delta/8\,. &(7)\end{aligned}$$

Also note that $P_{x_0 p_k} \subseteq Y_{j-1}$, so that $d_{Y_{j-1}}(x_j, p_k) = d_X(x_j, p_k)$, thus we conclude that

$$\begin{aligned} d_{X_j}(x_j, y) &\le d_{Y_{j-1}}(x_j, p_k) + d_{Y_{j-1}}(p_k, y)\\ &\overset{(4)}{\le} d_X(x_j, p_k) + 9\Delta/8 - d_X(x_0, p_k)\\ &= 9\Delta/8 - d_X(x_0, x_j) \overset{(3)}{\le} 9\Delta/8 - \Delta/4 = 7\Delta/8\,.\end{aligned}$$

□

Corollary 8. $y_1 \in X_0$.

Proof. Using Observation 5 we have that the radius of X_1 is at least $\Delta/8$, while the radius of any X_j with $j > 1$ is at most $\Delta/8$. Similarly to the proof of Claim 6 we have that all the t_j are of distance $\Delta/2$ from x_0. This suggests that $d_X(x_0, y_1) < 3\Delta/8$ and for $j > 1$ any point $u \in X_j$ satisfy $d_X(x_0, u) \ge 3\Delta/8$, so none of the X_j will contain y_1, thus $y_1 \in X_0$. □

Claim 9. $P_{x_0 t}(X) \subseteq G(X_0 \cup X_1)$.

Proof. If $t \in X_0$ then by Claim 1, $P_{x_0 t}(G(X)) \subseteq G(X_0)$. Otherwise, the choice of t_1 guarantees that $P_{x_0 t} \subseteq P_{x_0 t_1}$. Observe that the edge (y_1, x_1) lies on this path, which is decomposed into $P_{x_0 y_1}$ and $P_{x_1 t_1}$. By Corollary 8 $y_1 \in X_0$, so by Claim 1, $P_{x_0 y_1} \subseteq G(X_0)$, and also by Observation 3 $P_{x_1 t_1} \subseteq G(X_1)$. □

Claim 10. *When invoking* `hierarchical-petal-decomposition`$(G(X), x_0, t)$*, the only edges of* $G(X)$ *that are set to* $1/2$ *are those on* $P_{x_0 t}(G(X))$.

Proof. We will prove by induction on the depth of the recursion of `hierarchical-petal-decomposition`. The base case is trivial as V has x_0 as target. Assume by induction that X with center x_0 has a target t and only edges on $P_{x_0 t}$ are set to $1/2$. We partition X into $X_0, X_1, \ldots, X_s$, and we prove for these clusters.

- For X_0 with x_0 as center and target y_1, which was chosen on $P_{x_0 t_1}$. As $t \in P_{x_0 t_1}$ as well, and all the edges after y_1 are no longer in X_0, it must be that the edges set by X to $1/2$ are all on $P_{x_0 y_1}$.
- For X_1 with center x_1 and target t_1, which was chosen either as t or on a new path $(t = u_0, \ldots, u_l = t_1)$. As x_1 is on $P_{x_0 t_1}$ all the edges set by X to $1/2$ that are in X_1 are those on $P_{x_1 t}$. These edge are a subset of the edges X_1 is setting to $1/2$.
- For integer $j \ge 2$ and X_j, by Claim 9 all the edges set to $1/2$ by X lie in $G(X_0 \cup X_1)$, so X_j will contain only the edges that itself sets to $1/2$, which are on $P_{x_j t_j}$.

□

Claim 11. *The algorithm returns a tree.*

Proof. Assume by induction on the size of $G(X)$ that running `hierarchical-petal-decomposition` on $G(X)$ returns a tree. The base case is trivial for $|X| = 1$. Let $X \subseteq V$ be a cluster that is partitioned by `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 $|X| - 1$. 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 2 cycle edges leaving X_h. Observe that every edge (x_j, y_j) we added satisfies $y_j \in Y_j$, so $y_j \in X_k \cup X_0$ for some $k > j$. This means that only (x_h, y_h) can connect X_h to the other clusters in the cycle. All the other edges are either fully contained in some X_j, the (x_j, y_j) edges for $j > h$ cannot touch X_h, and for $j < h$, the minimality of h implies that there is no cycle edge touching X_j. □

4 Radius Bound

Let T be the tree created by calling `hierarchical-petal-decomposition` on G with center and target x_0, and let d_T be the shortest path metric in T, with respect to the *original* edge weights. Denote by $G^{(0)} = \{V\}$, and for integer $i \geq 1$, $G^{(i)}$ is the collection of clusters created from $G^{(i-1)}$ by performing `petal-decomposition` on every cluster of $G^{(i-1)}$, and applying the new edge weight of $1/2$ on the appropriate edges as defined in `hierarchical-petal-decomposition`. For any cluster $X \in G^{(i)}$ let $d_i = d_i(X)$ be the shortest path metric induced on $G(X)$. Since the clusters in $G^{(i)}$ are pairwise disjoint, we abuse notation and write only d_i (the cluster is inferred from context). We begin by showing that the shortest path from a center to its target and to the first petal always exists in the tree T.

Claim 12. *Fix some integer $i \geq 0$. Let $X \in G^{(i)}$ be a cluster with center x_0 and target t, then the following holds*

$$d_T(x_0, y_1) \leq 2d_i(x_0, y_1) \; . \tag{8}$$

Proof. We prove by induction on i that

- $d_T(x_0, y_1) \leq 2d_i(x_0, y_1)$,
- $d_T(x_0, t) = 2d_i(x_0, t)$.

First we prove the second bullet: by the induction hypothesis on $X_0 \in G^{(i+1)}$ with center x_0 and target which is by construction y_1, $d_T(x_0, y_1) = 2d_{i+1}(x_0, y_1) \leq 2d_i(x_0, y_1)$, where the last inequality holds because $P_{x_0 y_1}(X) \subseteq G(X_0)$ (using Corollary 8 to see that $y_1 \in X_0$ then by Claim 1), and in d_{i+1} we may set additional edges on this path to $1/2$.

Next we prove the second bullet. If it is the case that $t \in X_0$, then by Claim 1 also $P_{x_0 t}(X) \subseteq G(X_0)$. When forming X_1 we added a new path $P = (t = u_0, \ldots, u_l)$, and since $t \in X_0$ it must be that $(y_1, x_1) \in P$, so $t \in P_{x_0 y_1}$. By induction on X_0 with center x_0 and target y_1, we get that $d_T(x_0, y_1) = 2d_{i+1}(x_0, y_1)$. Since t lies on this shortest path, also

$$d_T(x_0, t) = 2d_{i+1}(x_0, t) = 2d_i(x_0, t) \; ,$$

where the last equality holds because edges on P_{x_0t} are already set to $1/2$ in d_i. By Claim 9 the only other case is that $t \in X_1$, in which case we have as above that $P_{x_0y_1}(X) \subseteq G(X_0)$ and by Observation 3 also $P_{x_1t}(X) \subseteq G(X_1)$. By applying induction on X_0 (with center x_0 and target y_1) and on X_1 (with center x_1 and target t_1, where $t \in P_{x_1t_1}$), and noting that $(y_1, x_1) \in P_{x_0t}$ was added to T, we get that

$$d_T(x_0,t) = d_T(x_0,y_1)+d_T(y_1,x_1)+d_T(x_1,t) = 2d_{i+1}(x_0,y_1)+2d_i(y_1,x_1)+2d_{i+1}(x_1,t) = 2d_i(x_0,t)\,.$$

□

Lemma 13. *For any $i \geq 1$ and any cluster $X \in G^{(i)}$,*

$$\mathrm{rad}(T[X]) \leq 4\mathrm{rad}(X)\,.$$

Proof. It suffices to prove by induction on i that for any cluster $X \in G^{(i)}$ with center x_0 and target t, and for any $y \in X$

$$d_T(x_0,y) \leq 4d_i(x_0,y)\,. \tag{9}$$

Assume X is partitioned into clusters $X_0, X_1, \ldots, X_s$. There are three cases to consider: $y \in X_0$, $y \in X_1$ and $y \in X_j$ with $j > 1$. 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 first portal, thus the path from x_0 to the first portal will surely appear in the tree Claim 12. The third case crucially uses the definition of petals and the re-weighting of the highways. For every point y in a petal, the re-weighting 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 with the metric $d_{i+1} = d_{i+1}(X_0)$ we obtain that $d_T(x_0,y) \leq 4d_{i+1}(x_0,y) \leq 4d_i(x_0,y)$. The last inequality holds since the shortest path $P_{x_0y}(X_0)$ can be the same as $P_{x_0y}(X)$, and we might have made some edges even shorter. This concludes the first case.

In the other two cases $y \in X_j$ for some $j \geq 1$. 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 some $j \geq 1$. For every $0 \leq \ell \leq r_j$ define $p_\ell = p_\ell^{(j)} \in P_{x_jt_j}$ be the point of distance ℓ from t_j. From here on fix any $0 \leq k \leq r_j$ 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 unique). By definition of a ball in a cone-metric

$$d_i(x_0,p_k) + d_{Y_{j-1}}(p_k,y) \leq d_i(x_0,y) + (r_j-k)/2\,, \tag{10}$$

we may write d_i instead of $d_{Y_{j-1}}$ because by Claim 1, $d_i(x_0,z) = d_{Y_{j-1}}(x_0,z)$ for all $z \in Y_{j-1}$. We shall use the following observations:

$$d_{i+1}(p_k,y) \leq d_{Y_{j-1}}(p_k,y) \tag{11}$$

$$d_i(x_0,x_j) + d_i(x_j,p_k) = d_i(x_0,p_k)\,. \tag{12}$$

To see (11), 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, $P_{p_ky}(Y_{j-1})$. The inequality follows because distances in X_j can only be made shorter due to re-weighting. For (12), this is simply because x_j is p_{r_j}, and all p_k are on the shortest path from t_j to x_0.

Case 2: $y \in X_1$. In this case we have the following

$$d_{i+1}(x_1, p_k) \le d_i(x_1, p_k) \ , \tag{13}$$

because $x_1 \in P_{x_0 p_k}(X)$, by Observation 3 we have that $P_{x_1 p_k}(X)$ is fully contained in X_1. The inequality follows because distances in X_1 can only be made shorter due to re-weighting.

By Claim 6 it follows that

$$2d_i(x_0, x_1) \ge \Delta/2 \tag{14}$$

By Observation 5 we have that

$$2(r_1 - k) \le 2r_1 \le \Delta/2 \ . \tag{15}$$

Recall that $Y_{j-1} = Y_0 = X$, hence $d_{Y_{j-1}} = d_i$. By the induction hypothesis on X_1,

$$\begin{aligned}
d_T(x_0, y) &\le d_T(x_0, y_1) + d_T(y_1, x_1) + d_T(x_1, y) \\
&\overset{(8)\wedge(9)}{\le} 2d_i(x_0, y_1) + 2d_i(y_1, x_1) + 4d_{i+1}(x_1, y) \\
&\le 2d_i(x_0, x_1) + 4d_{i+1}(x_1, p_k) + 4d_{i+1}(p_k, y) \\
&\overset{(11)\wedge(13)}{\le} 4(d_i(x_0, x_1) + d_i(x_1, p_k) + d_i(p_k, y)) - 2d_i(x_0, x_1) \\
&\overset{(12)}{=} 4(d_i(x_0, p_k) + d_i(p_k, y)) - 2d_i(x_0, x_1) \\
&\overset{(10)}{\le} 4(d_i(x_0, y) + (r_1 - k)/2) - 2d_i(x_0, x_1) \\
&\overset{(14)\wedge(15)}{\le} 4d_i(x_0, y) \ .
\end{aligned}$$

This concludes the proof for the second case.

Case 3: 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 W is a tree. Let X_0 be the root of the tree, and let $\mathrm{rank}(X_h)$ denote the depth of X_h in W. Observe that in the case $j \ge 2$, we have the following

$$d_i(x_j, p_k) = 2d_{i+1}(x_j, p_k) \ , \tag{16}$$

this holds because when $j > 1$, Claim 9 and Claim 10 suggests that the edges along $P_{x_j t_j}$ were not set to $1/2$ in d_i, so by Observation 3 the shortest path $P_{x_j t_j} \subseteq X_j$, and when these edges are set to $1/2$ in d_{i+1}, we reduce the shortest path distance by a factor of 2.

We will prove (9) in the case $y \in X_j$, $j \ge 2$, by induction on $\mathrm{rank}(X_j)$. The base case is when the rank is 1 and then it must be that $y_j \in X_0$. In this case by Claim 1 $P_{x_0 y_j} \subseteq G(X_0)$, so

$$d_{i+1}(x_0, y_j) \le d_i(x_0, y_j) \ . \tag{17}$$

By the induction hypothesis of (9) on both X_0 and X_j,

$$\begin{aligned}
d_T(x_0, y) &\leq d_T(x_0, y_j) + d_T(y_j, x_j) + d_T(x_j, y) \qquad (18)\\
&\overset{(9)}{\leq} 4d_{i+1}(x_0, y_j) + 2d_i(y_j, x_j) + 4d_{i+1}(x_j, y)\\
&\overset{(17)}{\leq} 4d_i(x_0, y_j) + 4d_i(y_j, x_j) + 4d_{i+1}(x_j, p_k) + 4d_{i+1}(p_k, y)\\
&\overset{(16)\wedge(11)}{\leq} 4d_i(x_0, x_j) + 2d_i(x_j, p_k) + 4d_{Y_{j-1}}(p_k, y)\\
&\overset{(12)}{=} 4(d_i(x_0, p_k) + d_{Y_{j-1}}(p_k, y)) - 2d_i(x_j, p_k)\\
&\overset{(10)}{\leq} 4(d_i(x_0, y) + (r_j - k)/2) - 2(r_j - k)\\
&= 4d_i(x_0, y) .
\end{aligned}$$

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

$$d_T(x_0, y_j) \leq 4d_i(x_0, y_j) . \qquad (19)$$

And finally

$$\begin{aligned}
d_T(x_0, y) &\leq d_T(x_0, y_j) + d_T(y_j, x_j) + d_T(x_j, y) \qquad (20)\\
&\overset{(9)\wedge(19)}{\leq} 4d_i(x_0, y_j) + 2d_i(y_j, x_j) + 4d_{i+1}(x_j, y)\\
&\leq 4d_i(x_0, y_j) + 4d_i(y_j, x_j) + 4d_{i+1}(x_j, p_k) + 4d_{i+1}(p_k, y)\\
&\overset{(16)\wedge(11)}{\leq} 4d_i(x_0, x_j) + 2d_i(x_j, p_k) + 4d_{Y_{j-1}}(p_k, y)\\
&\overset{(12)}{=} 4(d_i(x_0, p_k) + d_{Y_{j-1}}(p_k, y)) - 2d_i(x_j, p_k)\\
&\overset{(10)}{\leq} 4(d_i(x_0, y) + (r_j - k)/2) - 2(r_j - k)\\
&= 4d_i(x_0, y) .
\end{aligned}$$

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 very similar to the proof of [EEST05], and we give the details for completeness. Consider a single run of the algorithm `create-petal` on input (X, Y, t, x_0, R). Let $1 \leq p \leq L$ be as in the algorithm, and let $a = (1 + (p - 1)/L) \cdot R/2$ and $b = (1 + p/L) \cdot R/2$.

Claim 14. *The algorithm* `create-petal` *finds* $r \in [a, b]$ *satisfying*

$$|\partial(W_r)| < |E(W_r)| \cdot \frac{8L \ln \chi}{R} . \qquad (21)$$

Proof. Assume w.l.o.g that $R/2$ is even. First let us observe a basic property of our partition scheme, that if for some edge $(u,v)\in E$, $u\in W_r$, then $v\in W_{r+2}$. This holds simply because increasing r by 2 increases the radius of each cone in the petal by 1, and since cones are concentric system (see [EEST05]), and u belongs to some cone, we have that v will be included in that cone as well. This property enables us to claim that $|E(W_{r+2})|\geq|E(W_r)|+|\partial(W_r)|$.

Assume by contradiction that there is no such $r\in[a,b]$, then for all $r\in[a,b-2]$,

$$|E(W_{r+2})|\geq|E(W_r)|+|\partial(W_r)|\geq|E(W_r)|\left(1+\frac{8L\ln\chi}{R}\right),$$

Recall that $\chi=\frac{|E(X)|}{|E(W_a)|}$, and note that since $R\geq 10L\log n$, $(8L\ln\chi)/R\leq 1$ and thus $\left(1+\frac{8L\ln\chi}{R}\right)\geq e^{(4L\ln\chi)/R}=\chi^{4L/R}$. Now

$$\begin{aligned}
|E(W_b)| &\geq |E(W_{b-2})|\left(1+\frac{8\ln\chi}{R}\right)\geq|E(W_{b-4})|\left(1+\frac{8\ln\chi}{R}\right)^2\\
&\geq \cdots\geq|E(W_a)|\left(1+\frac{8\ln\chi}{R}\right)^{(b-a)/2}\\
&\geq |E(W_a)|\cdot\chi^{4L/R\cdot(b-a)/2}\\
&= |E(X)|\,,
\end{aligned}$$

but this is a contradiction. □

Consider now the algorithm `petal-decomposition` invoked on $G(X)$ with center x_0 and target t. It decomposes X into $X_0,X_1,\ldots,X_s$ (for some integer $s\geq 1$). For $j\in[s]$, let χ_j be the value defined at line 4. of `choose-radius` when creating the petal X_j, and denote by the $\mathrm{index}(X_j)$ the value of p chosen in line 3. By minimality of p, $|E(W_a)|\geq\frac{2|E(X)|}{2^{\log^{1-(p-1)/L}m}}$, so that

$$\ln\chi_j\leq 2\log^{1-(p-1)/L}m\leq 5\log^{1-p/L}m\,, \tag{22}$$

(the last inequality is because $\log^{1/L}m=2^{\log\log m/\log\log n}\leq 5/2$). Also observe that if some edge $(u,v)\in E$ is separated while decomposing the cluster X with radius Δ, then by Lemma 13

$$d_T(u,v)\leq 2\mathrm{rad}(T[X])\leq 8\Delta\,. \tag{23}$$

Let $\mathrm{avg\ stretch(BFS)}$ denote the total stretch over all clusters whose radius was smaller than $10\log n\log\log n$ and thus we created a BFS tree on them. Observe that $\mathrm{avg\ stretch(BFS)}=O(m\log n\log\log n)$, so this will add at most an additive factor to the total stretch, and we may ignore it. We now start to calculate the total stretch:

$$\begin{aligned}
\mathrm{TS}[X] &\overset{(23)}{\leq} \sum_{j=1}^{s}(\mathrm{TS}[X_j]+|\partial(X_j)|\cdot 8\Delta)\\
&\overset{(21)}{\leq} \sum_{j=1}^{s}\left(\mathrm{TS}[X_j]+8^3L|E(X_j)|\cdot\ln\chi_j\right)\\
&\overset{(22)}{\leq} \sum_{j=1}^{s}\mathrm{TS}[X_j]+O(L)\cdot\sum_{p=1}^{L}\sum_{j:\mathrm{index}(X_j)=p}|E(X_j)|\cdot\log^{1-p/L}m
\end{aligned} \tag{24}$$

Let us fix some edge $e \in E$, and analyze its contribution to (24). For every recursive level i in which $e \in E(X_j)$ with $p = \mathrm{index}(X_j)$ it contributed $O(L) \cdot \log^{1-p/L} m$. However by the choice of p, and by Claim 14 the radius r chosen for creating X_j satisfies $r \leq b$, so $|E(X_j)| \leq |E(W_b)| \leq \frac{2|E(X)|}{2^{\log^{1-p/L} m}}$. Intuitively, if p is small and thus the contribution is rather large, the size of the next cluster that contains e becomes much smaller, so e will participate in 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 is at most $O(L \cdot \log m) = O(\log \log \log n)$.

Formally, let $\ell_p(e)$ denote the number of recursive levels i in which e was in a cluster of index p. Then the number of edges in the clusters containing e decreased by a factor of at least $2^{\log^{1-p/L} m - 1}$, for every one of the $\ell_p(e)$ levels, so the total decrease is

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

because we started with m edges. This suggest that

$$\sum_{p=1}^{L} \ell_p(e)(\log^{1-p/L} m) \leq 2 \log m ,$$

where we used that $\sum_{p=1}^{L} \ell_p(e) \leq \log n \leq \log m$. Finally,

$$\mathrm{TS}[V] \leq O(L) \sum_{e \in E} \sum_{p=1}^{L} \ell_p(e) \log^{1-p/L} m = O(Lm \log m) = O(m \log n \log \log n) .$$

6 Extension to Weighted Graphs

Both papers [EEST05, ABN08] already showed how to deal with arbitrary weights on the edges. There are two ideas: the first is to contract edges shorter than $\mathrm{rad}(X)/n^2$, so that each edge participates in a logarithmic number of scales. The second is to add imaginary portal points when constructing cones, so that the algorithm is well defined. In our algorithm, in line 3.a of `hierarchical-petal-decomposition` we simply set the weight of edges to be $1/2$ of their original length, and observe that the analysis did not use the fact that edges have unit length.

For the analysis of stretch, the only real change is the proof of Claim 14, which still holds for weighted graphs, replacing $E(W_r)$ with $E^+(W_r)$ on the right hand side, see [EEST05] for a proof. This does not affect the total stretch by more than a factor of 2 because by (21), $2|E(W_r)| \geq |E^+(W_r)|$.

7 Fast Petal Construction

In order to bound the running time of our algorithm, we need to argue that the petal construction can be performed efficiently. It is shown in [KMPb] how to construct a star-decomposition on $G(X) = (V, E)$ 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 shortest path from the center of the cluster, 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 see that both cones and petals may be constructed efficiently, by a region growing scheme.

Given a weighted undirected graph $G = (V, E, w)$, let p_k be the point of distance k from t on the shortest path P_{tx} from t to x, and all distances d are with respect to G. Let $\tilde{G} = (V, A, w')$ be the weighted directed graph induced by adding the two edges $(u \to v), (v \to u) \in A$ for each $(u, v) \in E$, and setting $w'(u \to v) = d(u, v) - (d(v, x) - d(u, x))$. 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 $(p_i \to p_{i+1})$ is changed to $w(p_i, p_{i+1})/2$ for all $i < r$. Recall that the petal with center x, target t and radius r was defined in the algorithm as
$W_r = \bigcup_{p \in P_{xt} \,:\, d(p,t) \le r} B_{(V,\rho(V,x,p))}(p, (r - d(p, t))/2)$.

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

Proof. First we prove that for any $r \ge 0$, $W_r \subseteq P(x, t, r)$. Fix some $v \in W_r$, and let $0 \le k \le r$ be such that $v \in B_{(V,\rho(V,x,p_k))}(p_k, (r - k)/2)$. Observe that by the re-weighting 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 $d(v, p_k)+d(p_k, x) \le d(v, x)+(r-k)/2$. 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 w' it follows that

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

as required.

Let $0 = r_1 < r_2 < \cdots < r_k$ be all the possible radii for which the size of $P(x, t, r)$ changes. We prove that $P(x, t, r) \subseteq W_r$ by induction on the radius. The base case for $r_1 = 0$, then $W_0 = \{y \,:\, d(y, x) = d(y, t) + d(t, x)\}$, and $P(x, t, 0)$ will contain all points reachable with 0 weight edges, by definition these edges $(u \to v)$ are the ones that satisfy $d(v, x) - d(u, x) = d(u, v)$, so any path leaving t using these edges will lead to a point y for which $d(y, x) = d(y, t) + d(t, x)$.

For the inductive step, assume $P(x, t, r_{i-1}) \subseteq W_{r_{i-1}}$, and prove for r_i. Let $\delta = r_i - r_{i-1}$. Let $v \in P(x, t, r_i) \setminus P(x, t, r_{i-1})$, and assume $u \in P(x, t, r_{i-1})$ is such that $(u \to v) \in A$ with $w'(u \to v) \le \delta/2$. Then by definition of w' we have that $d(u, v) \le \delta + d(v, x) - d(u, x)$. By the induction hypothesis we have that $u \in W_{r_{i-1}}$, so let k be such that $u \in B_{(V,\rho(V,x,p_k))}(p_k, (r_{i-1} - k)/2)$, by definition of cone metric $d(u, p_k) + d(p_k, x) \le d(u, x) + (r_{i-1} - k)/2$. It follows that

$$\begin{aligned} d(v, p_k) + d(p_k, x) &\le d(v, u) + d(u, p_k) + d(p_k, x) \\ &\le \delta + d(v, x) - d(u, x) + d(u, x) + (r_{i-1} - k)/2 \\ &= d(v, x) + (r_i - k)/2 \,, \end{aligned}$$

meaning that $v \in W_{r_i}$. □

Acknowledgments

We would like to thank Yair Bartal, Kunal Talwar and Michael Elkin for helpful discussions.

References

[ABN06] Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, STOC '06, pages 271–286, New York, NY, USA, 2006. ACM.

[ABN07] Ittai Abraham, Yair Bartal, and Ofer 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, pages 502–511, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[ABN08] Ittai Abraham, Yair Bartal, and Ofer Neiman. Nearly tight low stretch spanning trees. In *FOCS '08: Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 781–790, Washington, DC, USA, 2008. IEEE Computer Society.

[AKPW95] Noga Alon, Richard M. Karp, David Peleg, and Douglas West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, 1995.

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

[Bar98] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 161–168, New York, NY, USA, 1998. ACM.

[Bar04] Yair Bartal. Graph decomposition lemmas and their role in metric embedding methods. In Susanne Albers and Tomasz Radzik, editors, *ESA*, volume 3221 of *Lecture Notes in Computer Science*, pages 89–97. Springer, 2004.

[BIS07] Mihai Bădoiu, Piotr Indyk, and Anastasios Sidiropoulos. Approximation algorithms for embedding general metrics into trees. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 512–521, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[CCGG98] Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto 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*, pages 114–123, New York, NY, USA, 1998. ACM Press.

[CDN+10] Victor Chepoi, Feodor F. Dragan, Ilan Newman, Yuri Rabinovich, and Yann Vaxès. Constant approximation algorithms for embedding graph metrics into trees and outerplanar graphs. In *Proceedings of the 13th international conference on Approximation, and 14 the International conference on Randomization, and combinatorial optimization: algorithms and techniques*, APPROX/RANDOM'10, pages 95–109, Berlin, Heidelberg, 2010. Springer-Verlag.

[CKM+11] Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 273–282, New York, NY, USA, 2011. ACM.

[EEST05] Michael Elkin, Yuval Emek, Daniel A. Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 494–503, New York, NY, USA, 2005. ACM Press.

[EP04] Yuval Emek and David Peleg. Approximating minimum max-stretch spanning trees on unweighted graphs. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '04, pages 261–270, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[EP06] Yuval Emek and David 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, pages 1045–1053, New York, NY, USA, 2006. ACM.

[FRT03] Jittat Fakcharoenphol, Satish Rao, and Kunal 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, pages 448–455, New York, NY, USA, 2003. ACM.

[KMPa] Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching optimality for solving sdd linear systems. In *51th Annual IEEE Symposium on Foundations of Computer Science, October 23-26, 2010*, pages 235–244, Las Vegas, Nevada, USA.

[KMPb] Ioannis Koutis, Gary L. Miller, and Richard Peng. A nearly $O(m \log n)$ time solver for SDD linear systems. In *52th Annual IEEE Symposium on Foundations of Computer Science, 2011*.

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

[Sey95] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, June 1995.

[SS08] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 563–568, New York, NY, USA, 2008. ACM.

[ST04] Daniel A. Spielman and Shang-Hua 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, pages 81–90, New York, NY, USA, 2004. ACM.