Approximation Algorithms for Embedding General Metrics Into Trees *

Mihai Bădoiu† Piotr Indyk† Anastasios Sidiropoulos†

Abstract

We consider the problem of embedding general metrics into trees. We give the first non-trivial approximation algorithm for minimizing the multiplicative distortion. Our algorithm produces an embedding with distortion $(c \log n)^{O(\sqrt{\log \Delta})}$, where c is the optimal distortion, and Δ is the spread of the metric (i.e. the ratio of the diameter over the minimum distance). We give an improved $O(1)$-approximation algorithm for the case where the input is the shortest path metric over an unweighted graph. Moreover, we show that by composing our approximation algorithm for embedding general metrics into trees, with the approximation algorithm of [BCIS05] for embedding trees into the line, we obtain an improved approximation algorithm for embedding general metrics into the line.

We also provide almost tight bounds for the relation between embedding into trees and embedding into spanning subtrees. We show that for any unweighted graph G, the ratio of the distortion required to embed G into a spanning subtree, over the distortion of an optimal tree embedding of G, is at most $O(\log n)$. We complement this bound by exhibiting a family of graphs for which the ratio is $\Omega(\log n/ \log \log n)$.

1 Introduction

A low-distortion embedding between two metric spaces M and M' with distance functions D and D' is a (non-contractive) mapping f such that for any pair of points p, q in the original metric, their distance $D(p, q)$ before the mapping is the same as the distance $D'(f(p), f(q))$ after the mapping, up to a (small) multiplicative factor c. Low-distortion embeddings have been a subject of extensive mathematical studies, and found numerous applications in computer science (cf. [Lin02, Ind01]).

More recently, a few papers (cf. Figure 1) addressed the *relative* or *approximation* version of this problem. In this setting, the question is: for a class of metrics C, and a host metric M', what is the *smallest approximation factor* $a \geq 1$ of an efficient[1] algorithm minimizing the distortion of embedding of a given input metric $M \in C$ into M' ? This formulation enables the algorithm to adapt to a given input metric. In particular, if the host metric is "expressive enough" to accurately model the input distances, the minimum achievable distortion is low, and the algorithm will produce an embedding with low distortion as well.

This problem has been a subject of extensive applied research during the last few decades (e.g., see [MDS] web page, or [KTT98]). However, almost all known algorithms for this problem are heuristic. As such, they can get stuck in local minima, and do not provide any global guarantees on solution quality ([KTT98], section 2).

In this paper we consider the problem of approximating minimum distortion for embedding general metrics into *tree metrics*, i.e., shortest path metric over (weighted) trees. This is a natural problem with connections and applications to many areas. The classic application is the recovery of evolutionary trees from evolutionary distances between the data (e.g., see [Sci05], or [DEKM98], section 7.3). Another motivation comes from computational geometry. Specifically, Eppstein ([Epp00], Open Problem 4) posed a question about algorithmic complexity of finding the *minimum-dilation spanning tree* of a given set of points in the plane. This problem is equivalent (up to a constant factor in the approximation factor) to a special case of our problem, where the input metric is induced by points in the plane. Moreover, a closely related problem has been studied in the context of graph spanners [PU87, PR98]. Namely, the problem of computing a *minimum-stretch spanning tree* of a graph can be phrased as the problem of computing the minimum distortion embedding of a graph into a spanning subtree.

1.1 Our results Our main results are the first non-trivial approximation algorithms for embedding into tree metrics, for minimizing the multiplicative distortion. Specifically, if the input metric is an unweighted

*This work was supported in part by NSF CAREER award CCR-0133849, David and Lucille Packard Fellowship and Alfred P. Sloan Fellowship.

†MIT Computer Science and Artificial Intelligence Laboratory; Stata Center, Room 32-G596; Cambridge, Massachusetts 02139; {mihai,indyk,tasos}@theory.lcs.mit.edu

[1] That is, with running time polynomial in n, where n is the number of points of the metric space.

graph, we give a $O(1)$-approximation algorithm for this problem. For general metrics, we give an algorithm such that if the input metric is c-embeddable into some tree metric, produces an embedding with distortion $\alpha(c \log n)^{O(\log_\alpha \Delta)}$, for any $\alpha \geq 1$. In particular, by setting $\alpha = 2^{\sqrt{\log \Delta}}$, we obtain distortion $(c \log n)^{O(\sqrt{\log \Delta})}$. Alternatively, when $\Delta = n^{O(1)}$, by setting $\alpha = n^\epsilon$, we obtain distortion $n^\epsilon (c \log n)^{O(1/\epsilon)}$. This in turn yields an $O(n^{1-\beta})$-approximation for some $\beta > 0$, since it is always possible to construct an embedding with distortion $O(n)$ in polynomial time [Mat90].

Further, we show that by composing our approximation algorithm for embedding general metrics into trees, with the approximation algorithm of [BCIS05] for embedding trees into the line, we obtain an improved [2] approximation algorithm for embedding general metrics into the line. The best known distortion guarantee for this problem [BCIS05] was $c^{O(1)}\Delta^{3/4}$, while the composition results in distortion $(c \log n)^{O(\sqrt{\log \Delta})}$. In fact, we provide a general framework for composing relative embeddings which could be useful elsewhere.

For the special case where the input is an unweighted graph metric, we also study the relation between embedding into trees, and embedding into spanning subtrees. An $O(\log n)$-approximation algorithm is known [EP04] for this problem. We show that if an unweighted graph metric embeds into a tree with distortion c, then it also embeds into a spanning subtree with distortion $O(c \log n)$. We also exhibit an infinite family of graphs that almost achieves this bound; each graph in the family embeds into a tree with distortion $O(\log n)$, while any embedding into a spanning subtree has distortion $\Omega(\log^2 n / \log \log n)$. We remark that by composing the upper bound with our $O(1)$-approximation algorithm for unweighted graphs, we recover the result of [EP04]. Due to lack of space, we defer the results on the relation between embedding into trees, and embedding into spanning subtrees, to the full version of this paper.

1.2 Related Work

The study of the problem of approximating metrics by tree metrics has been initiated in [FCKW93, ABFC$^+$96], where the authors give an $O(1)$-approximation algorithm for embedding metrics into tree metrics. They also provide exact algorithms for embeddings into simpler metrics, called *ultrametrics*. However, instead of the *multiplicative* distortion (defined as above), their algorithms optimize the *additive* distortion; that is, the quantity $\max_{p,q} |D(p,q) - D'(p,q)|$. The same problem has recently been studied also for the case of minimizing the L_p norm of the differences [HKM05, AC05]. In a recent paper [AC05], a $(\log n \log \log n)^{1/p}$-approximation has been obtained for this problem.

Minimizing the multiplicative distortion seems to be a harder problem in general. For example, embedding into the line is hard to $n^{\Omega(1)}$-approximate for multiplicative distortion, and there is no known poly(c)-approximation algorithm, while for additive distortion there exists a simple 3-approximation.

The problem of embedding into a tree with minimum multiplicative distortion is closely related to the problem of computing a minimum-stretch spanning tree. The two problems are identical for the case of complete graphs. We mention the work of [PU87, CC95, VRM$^+$97, PR98, PT01, FK01, EP04]. For unweighted graphs, the best known approximation is an $O(\log n)$-approximation algorithm [EP04]. Our algorithm for unweighted graphs can be combined with our algorithm for converting an embedding into a tree into an embedding into a spanning subtree, to give the same approximation guarantee (within constant factors).

The problem of approximating the *multiplicative* distortion of embeddings into *ultrametrics* has been studied as well; there is a polynomial-time algorithm for solving this problem exactly [ABD$^+$05]. Ultrametrics are useful for modeling evolutionary data, but they are not as expressive as general tree metrics. In particular, they form a proper subset of tree metrics. See [DEKM98] for a more detailed discussion.

1.3 Notation and Definitions

Graphs For a graph $G = (V, G)$, and $U \subseteq V(G)$, let $G[U]$ denote the subgraph of G induced by U. For $u, v \in V(G)$ let $D_G(u, v)$ denote the shortest-path distance between u and v in G. We assume that all the edges of G have weight at least 1. If G is weighted let W_G denote the maximum edge weight of G, and let $W_G = 1$ otherwise.

Metrics For any finite metric space $M = (X, D)$, we assume that the minimum distance in M is at least 1. M is called a *tree metric* iff it is the shortest-path metric of a subset of the vertices of a weighted tree. For a graph $G = (V, E)$, and $\gamma \geq 1$ we say that G γ-approximates M if $V(G) \subseteq X$, and for each $u, v \in V(G)$, $D(u, v) \leq D_G(u, v) \leq \gamma D(u, v)$. We say that M c-embeds into a tree if there exists an embedding of M into a tree with distortion at most c. When considering an embedding into a tree, we assume unless stateted otherwise that the tree might contain steiner nodes. By a result of Gupta [Gup01], after computing

[2] Strictly speaking, the guarantees are incomparable, but the dependence on Δ in our algorithm is a great improvement over the earlier bound.

Paper	From	Into	Distortion	Comments
[LLR94]	general metrics	L_2	c	uses SDP
[KRS04]	line	line	c	c is constant, embedding is a bijection
	unweighted graphs	bounded degree trees	c	c is constant, embedding is a bijection
[PS05]	$\Re^3$	$\Re^3$	$> (3-\epsilon)c$	hard to 3-approximate, embedding is a bijection
[HP05]	line	line	$> n^{\Omega(1)}$	$c = n^{\Omega(1)}$, embedding is a bijection
[EP04]	unweighted graphs	sub-trees	$O(c \log n)$	
[PT01]	outerplanar graphs	sub-trees	c	
[CC95]	unweighted graphs	sub-trees	NP-complete	
[FK01]	planar graphs	sub-trees	NP-complete	
[BDG+05]	unweighted graphs	line	$O(c^2)$	implies $\sqrt{n}$-approximation
			$> ac$	hard to a-approximate for some $a > 1$
			c	c is constant
	unweighted trees	line	$O(c^{3/2}\sqrt{\log c})$	
	subsets of a sphere	plane	$3c$	
[BCIS06]	ultrametrics	$\Re^d$	$c^{O(d)}$	
[ABD+05]	general metrics	ultrametrics	c	
[BCIS05]	general metrics	line	$O(\Delta^{3/4}c^{11/4})$	
	weighted trees	line	$c^{O(1)}$	
	weighted trees	line	$\Omega(n^{1/12}c)$	hard to $O(n^{1/12})$-approximate even for $\Delta = n^{O(1)}$
[LNP06]	weighted trees	L_p	$O(c)$	

Figure 1: Previous work on relative embedding problems for multiplicative distortion. We use c to denote the optimal distortion, and n to denote the number of points in in the input metric. Note that the table contains only the results that hold for the *multiplicative* definition of the distortion; there is a rich body of work that applies to other definitions of distortion, notably the *additive* or *average* distortion, see [BCIS05] for an overview.

the embedding we can remove the steiner nodes losing at most a $O(1)$ factor in the distortion (and thus also in the approximation factor).

α-Restricted Subgraphs For a weighted graph $G = (V, E)$, and for $\alpha > 0$, the *α-restricted subgraph of* G is defined as the graph obtained from G after removing all the edges of weight greater than α. Similarly, for a metric $M = (X, D)$, the α-restricted subgraph of M is defined as the weighted graph on vertex set X, where an edge $\{u, v\}$ appears in G iff $D(u, v) \leq \alpha$, and the weight of every edge $\{u, v\}$ is equal to $D(u, v)$.

2 A Forbidden-Structure Characterization of Tree-Embeddability

Before we describe our algorithms, we give a combinatorial characterization of graphs that embed into trees with small distortion. For any $c > 1$, the characterization defines a forbidden structure that cannot appear in a graph that embeds into a tree with distortion at most c. This structure will be later used when analyzing our algorithms to show that the computed embedding is close to optimal.

LEMMA 2.1. *Let $G = (V, E)$ be a (possibly weighted) graph. If there exist nodes $v_0, v_1, v_2, v_3 \in V(G)$, and $\lambda > 0$, such that*

- *for each $i : 0 \leq i < 4$, there exists a path p_i, with endpoints v_i, and $v_{i+1 mod 4}$, and*
- *for each $i : 0 \leq i < 4$, $D_G(p_i, p_{i+2 mod 4}) > \lambda W_G$,*

then, any embedding of G into a tree has distortion greater than λ.

Proof. Let $W = W_G$. Consider an optimal non-contracting embedding f of G, into a tree T. For any $u, v \in V(G)$, let $P_{u,v}$ denote the path from $f(u)$ to $f(v)$, in T. For each i, with $0 \leq i < 4$, define T_i as the minimum subtree of T, which contains all the images of the nodes of p_i. Since each T_i is minimum, it follows that all the leaves of T_i are nodes of $f(p_i)$.

CLAIM 1. *For each i, with $0 \leq i < 4$, we have $T_i = \bigcup_{\{u,v\} \in E(p_i)} P_{u,v}$.*

Proof. Assume that the assertion is not true. That is, there exists $x \in V(T_i)$, such that for any $\{u, v\} \in E(p_i)$, the path $P_{u,v}$ does not visit x. Clearly, $x \notin V(p_i)$, and thus x is not a leaf. Let $T_i^1, T_i^2, \ldots, T_i^j$, be the connected components obtained by removing x from T_i. Since for every $\{u, v\} \in E(p_i)$, $P_{u,v}$ does not visit x, it follows that there is no edge $\{u, v\} \in E(p_i)$, with $u \in T_i^a$, $v \in T_i^b$, and $a \neq b$. This however, implies that p_i is not connected, a contradiction. □

CLAIM 2. *For each i, with $0 \leq i < 4$, we have $T_i \cap T_{i+2 mod 4} = \emptyset$.*

Proof. Assume that the assertion does not hold. That is, there exists i, with $0 \leq i < 4$, such that $T_i \cap T_{i+2\text{mod}4} \neq \emptyset$. We have to consider the following two cases:

Case 1: *$T_i \cap T_{i+2mod4}$ contains a node from $V(p_i) \cup V(p_{i+2mod4})$.* W.l.o.g., we assume that there exists $w \in V(p_{i+2\text{mod}4})$, such that $w \in T_i \cap T_{i+2\text{mod}4}$. By Claim 1, it follows that there exists $\{u, v\} \in E(p_i)$, such that $f(w)$ lies on $P_{u,v}$. This implies $D_T(f(u), f(v)) = D_T(f(u), f(w)) + D_T(f(w), f(v))$. On the other hand, we have $D_G(p_i, p_{i+2\text{mod}4}) > \lambda W$, and since f is non-contracting, we obtain $D_T(f(u), f(v)) > 2\lambda W$. Thus, $c \geq D_T(f(u), f(v))/D_G(u, v)$. Since $\{u, v\} \in E(G)$, and the maximum edge weight in G is at most W, we have $D_G(u, v) \leq W$, and thus $c > 2\lambda$.

Case 2: *$T_i \cap T_{i+2mod4}$ does not contain nodes from $V(p_i) \cup V(p_{i+2mod4})$.* Let $w \in T_i \cap T_{i+2\text{mod}4}$. By Claim 1, there exist $\{u_1, v_1\} \in E(p_i)$, and $\{u_2, v_2\} \in E(p_{i+2\text{mod}4})$, such that w lies in both P_{u_1,v_1}, and P_{u_2,v_2}. We have $D_T(f(u_1), f(v_1)) + D_T(f(u_2), f(v_2)) = D_T(f(u_1), f(w)) + D_T(f(w), f(v_1)) + D_T(f(u_2), f(w)) + D_T(f(w), f(v_2)) \geq D_T(f(u_1), f(u_2)) + D_T(f(v_1), f(v_2)) \geq D_G(u_1, u_2) + D_G(v_1, v_2) \geq 2D_G(p_i, p_{i+2\text{mod}4}) > 2\lambda W$. Thus, we can assume that $D_T(f(u_1), f(v_1)) > \lambda W$. It follows that $c \geq \frac{D_T(f(u_1), f(v_1))}{D_G(u_1, v_1)} > \lambda$. □

Moreover, since p_i, and $p_{i+1\text{mod}4}$, share an endpoint, we have $T_i \cap T_{i+1\text{mod}4} \neq \emptyset$. By Claim 2, it follows, that $\bigcup_{i=0}^{3} T_i \subseteq T$, contains a cycle, a contradiction. □

3 Tree-Like Decompositions

In this section we describe a graph partitioning procedure which is a basic step in our algorithms. Intuitively, the procedure partitions a graph into a set of clusters, and arranges the clusters in a tree, so that the structure of the tree of clusters resembles the structure of the original graph.

Formally, the procedure takes as input a (possibly weighted) graph $G = (V, E)$, a vertex $r \in V(G)$, and a parameter $\lambda \geq 1$. The output of the procedure is a pair $(T_{\mathcal{K}}^G, \mathcal{K}_G)$, where $\mathcal{K}_G$ is a partition of $V(G)$, and $T_{\mathcal{K}}^G$ is a rooted tree with vertex set $\mathcal{K}_G$.

The partition $\mathcal{K}_G$ of $V(G)$ is defined as follows. For integer i, let

$$V_i = \{v \in V(G) | W_G(i-1)\lambda \leq D_G(r, v) < W_G i\lambda\}.$$

Initially, $\mathcal{K}_G$ is empty. Let t be the maximum index such that V_t is non-empty. Let $Y_i = \bigcup_{j=i}^{t} V_j$. For each $i \in [t]$, and for each connected component Z of $G[Y_i]$ that intersects V_i, we add the set $Z \cap V_i$, to the partition $\mathcal{K}_G$. Observe that some clusters in $\mathcal{K}_G$ might induce disconnected subgraphs in G.

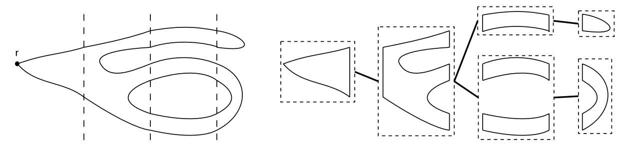

Figure 2: An example of a tree-like decomposition of a graph.

$T_{\mathcal{K}}^G$ can now be defined as follows. For each $K, K' \in \mathcal{K}_G$, we add the edge $\{K, K'\}$ in $T_{\mathcal{K}}^G$ iff there is an edge in G between a vertex in K and a vertex in K'. The root of $T_{\mathcal{K}}^G$ is the cluster containing r. The resulting pair $(T_{\mathcal{K}}^G, \mathcal{K}_G)$ is called a (r, λ)-*tree-like decomposition of* G.

Figure 2 depicts the described decomposition.

Proposition 3.1. *$T_{\mathcal{K}}^G$ is a tree.*

Proof. Let $u, v \in V(G)$. Since G is connected, there is a path p from u to v in G. Let $p = x_1, \ldots, x_{|p|}$. For each $i \in \{1, \ldots, |p|\}$, let $K_i \in \mathcal{K}_G$ be such that $x_i \in K_i$. It is easy to verify that the sequence $\{K_i\}_{i=1}^{|p|}$ contains a sub-sequence that corresponds to a path in $T_{\mathcal{K}}^G$. Thus, $T_{\mathcal{K}}^G$ is connected.

It is easy to show by induction on i that for $i = t, \ldots, 1$, the subset $L_i \subseteq \mathcal{K}_G$ that is obtained by partitioning $\bigcup_{j=i}^{t} V_j$, induce a forest in $T_{\mathcal{K}}^G$. Since $L_1 = \mathcal{K}_G$, and $T_{\mathcal{K}}^G$ is connected, it follows that $T_{\mathcal{K}}^G$ is a tree. □

3.1 Properties of Tree-Like Decompositions

Before using the tree-like decompositions in our algorithms, we will show that for a certain range of the decomposition parameters, they exhibit some usefull properties.

We will first bound the diameter of the clusters in $\mathcal{K}_G$. The intuition behind the proof is as follows. If a cluster K is long enough, then starting from a pair of vertices in $x, y \in K$ that are far from each other, and tracing the shortest paths from x and y to r, we can discover the forbidden structure of lemma 2.1 in G. Applying lemma 2.1 we obtain a lower bound on the optimal distortion, contradicting the fact that G embeds into a tree with small distortion.

Lemma 3.1. *Let $G = (V, E)$ be a graph that γ-embeds into a tree, let $r \in V(G)$, and let $(T_{\mathcal{K}}^G, \mathcal{K}_G)$ be a (r, γ)-tree-like decomposition of G. Then, for any $K \in \mathcal{K}_G$, and for any $u, v \in K$, $D_G(u, v) \leq 20\gamma W_G$.*

Proof. Assume that the assertion is not true, and pick $K \in \mathcal{K}_G$, and vertices $x, y \in K$, such that $D_G(x, y) > 20\gamma W_G$. Recall that $\mathcal{K}_G$ was obtained by partitioning the vertices of G according to their distance from r. Let q_x, and q_y be the shortest paths from x to r, and from y to r respectively. Let $K_1, \ldots, K_\tau$ be the branch in $T_{\mathcal{K}}^G$, such that $r \in K_1$, and $K_\tau = K$. By the construction of $\mathcal{K}_G$, we have that for any $i \in [\tau]$, for any $z \in K_i$, $D_G(r, z) \leq iW_G\gamma$. Thus, $D_G(x, y) \leq D_G(x, r) + D_G(r, y) \leq 2\tau W_G c$. Since $D_G(x, y) > 20\gamma W_G$, it follows that $\tau > 10$.

Consider now the sub-path p^x of q_x that starts from x, and terminates to the first vertex x' of $K_{\tau-2}$ visited by q_x. Define similarly p^y as the sub-path of q_y that starts from y, and terminates to the first vertex y' of $K_{\tau-2}$ visited by q_y. We will first show that $D_G(p^x, p^y) > \gamma W_G$. Observe that by the construction of $\mathcal{K}_G$, we have that $D_G(x, x') \leq 2\gamma W_G$, and also $D_G(y, y') \leq 2\gamma W_G$. Since p^x, and p^y are shortest paths, we have that for any $z \in p^x$, $D_G(x, z) \leq 2\gamma W_G$, and similarly for any $z \in p^y$, $D_G(y, z) \leq 2\gamma W_G$. Pick $z \in p^x$, and $z' \in p^y$, such that $D_G(z, z')$ is minimized. We have $D_G(x, y) \leq D_G(x, z) + D_G(z, z') + D_G(z', y) \leq D_G(z, z') + 4\gamma W_G$. Thus, $D_G(p^x, p^y) = D_G(z, z') \geq D_G(x, y) - 4\gamma W_G > 20\gamma W_G - 4\gamma W_G = 16\gamma W_G$.

Let now $p^{x\prime}$ be the remaining sub-path of q_x, starting from x', and terminating to r, and define $p^{y\prime}$ similarly. Let p^{xy} be the path from x' to y', obtained by concatenating $p^{x\prime}$, and $p^{y\prime}$.

By the construction of $\mathcal{K}_G$ it follows that if we remove from G all the vertices in the sets $K_1, K_3, \ldots, K_{\tau-1}$, then x and y remain in the same connected component. In other words, we can pick a path p^{yx} from x to y, that does not visit any of the vertices in $\bigcup_{j=1}^{\tau-1} K_j$. It follows that the distance between any vertex of p^{yx}, and any vertex in $\bigcup_{j=1}^{\tau-2} K_j$, is greater than γW_G. Thus, $D_G(p^{xy}, p^{yx}) > \gamma W_G$.

We have thus shown that there are vertices $x, y, y', x' \in V(G)$, and paths p^x, p^y, p^{xy}, p^{yx}, satisfying the conditions of Lemma 2.1. It follows that the optimal distortion required to embed G into a tree is greater than γ, a contradiction. □

Using the bound on the diameter of the clusters in $\mathcal{K}_G$, we can show that for certain values of the parameters, the distances in the tree of clusters approximate the distances in the original graph.

LEMMA 3.2. *Let $G = (V, E)$ be a graph that γ-embeds into a tree, let $r \in V(G)$, and let $(T_{\mathcal{K}}^G, \mathcal{K}_G)$ be a (r, γ)-tree-like decomposition of G. Then, for any $K_1, K_2 \in \mathcal{K}_G$, and for any $x_1 \in K_1$, $x_2 \in K_2$, $(D_{T_{\mathcal{K}}^G}(K_1, K_2) - 2)W_G\gamma \leq D_G(x_1, x_2) \leq (D_{T_{\mathcal{K}}^G}(K_1, K_2) + 2)20W_G\gamma$.*

Proof. Let $\delta = D_{T_{\mathcal{K}}^G}(K_1, K_2)$. We begin by showing the first inequality. We have to consider the following cases:

Case 1: *K_1 and K_2 are on the same path from the root to a leaf of $T_{\mathcal{K}}^G$.* Let the path between K_1 and K_2 in $T_{\mathcal{K}}^G$ be $K_1, H_1, H_2, \ldots, H_{\delta-1}, K_2$. Assume that the assertion is not true. That is, $D_G(x_1, x_2) < (\delta - 2)W_G\gamma$. Thus, $D_G(r, x_2) \leq D_G(r, x_1) + D_G(x_1, x_2) < D_G(r, x_1) + (\delta - 1)W_G\gamma$. Assume that $r \in K_r$, for some $K_r \in \mathcal{K}_G$, and w.l.o.g. that K_1 is an ancestor of K_2 in $T_{\mathcal{K}}^G$. Let the distance between K_r and K_1 in $T_{\mathcal{K}}^G$ be k. Then, the distance between K_r and K_2 is at most $k' = k + D_G(x_1, x_2)/(W_G\gamma)$. This implies that $\delta = k' - k < \delta - 1$, a contradiction.

Case 2: *K_1 and K_2 are not on the same path from the root to a leaf of $T_{\mathcal{K}}^G$.* Let K_a be the nearest common ancestor of K_1 and K_2 in $T_{\mathcal{K}}^G$. Observe that any path from x to y in G passes through K_a. Thus, we have $D_G(x, y) \geq D_G(K_x, K_a) + D_G(K_a, K_y)$. Let δ_i, for $i \in \{1, 2\}$ be the distance between K_a and K_i in $T_{\mathcal{K}}^G$. Then, by an argument similar to the above, we obtain that $D_G(K_x, K_a) \geq (\delta_1 - 1)W_G\gamma$, and also $D_G(K_y, K_a) \geq (\delta_2 - 1)W_G\gamma$. Since K_a is the nearest common ansestor of K_1 and K_2, it follows that K_a reparates K_1 from K_2 in G. Thus, $D_G(x, y) \geq D_G(K_x, K_y) \geq D_G(K_x, K_a) + D_G(K_y, K_a) \geq (\delta - 2)W_G c$.

We now show the second inequality. Consider an edge $\{K, K'\}$ of $T_{\mathcal{K}}^G$. Since K and K' are connected in $T_{\mathcal{K}}^G$ it follows that there exists an edge in G between a vertex in K and a vertex in K'. Since the maximum edge weight of G is W_G, we obtain $D_G(K, K') \leq W_G$.

Since by Lemma 3.1, the diameter of each $K \in \mathcal{K}_G$ is at most $20W_G\gamma$, it follows that $D_G(x_1, x_2) \leq \delta W_G + (\delta + 1)20W_G\gamma < (\delta + 2)20W_G\gamma$. □

4 Approximation Algorithm for Embedding Unweighted Graphs

In this section we give a $O(1)$-approximation algorithm for the problem of embedding the shortest path metric of an unweighted graph into a tree. Informally, the algorithm works as follows. Let $G = (V, E)$ be an unweighted graph, such that G can be embedded into an unweighted tree with distortion c. At a first step, we compute a tree-like decomposition $(T_{\mathcal{K}}^G, \mathcal{K}_G)$ of G. For each cluster in $\mathcal{K}_G$ we embed the vertices of the cluster in a star. We then connect the starts to form a tree embedding of G by connecting stars that correspond to clusters that are adjancent in $T_{\mathcal{K}}^G$.

Formally, the algorithm can be described with the following steps.

Step 1. We pick $r \in V(G)$, and we compute a (r, c)-tree-like decomposition $(T_{\mathcal{K}}^G, \mathcal{K}_G)$ of G.

Step 2. We construct a tree T as follows. Let $\mathcal{K}_G =$

$\{K_1, \ldots, K_t\}$. For each $i \in [t]$, we construct a star with center a new vertex ρ_i, and leaves the vertices in K_i. Next, for each edge $\{K_i, K_j\}$ in $T_{\mathcal{K}}^G$, we add an edge $\{\rho_i, \rho_j\}$ in T.

By proposition 3.1, we know that the resulting graph T is indeed a tree, so we can focus of bounding the distortion of T. By lemma 3.1, the diameter of each cluster in $\mathcal{K}_G$ is at most $20cW_G = 20c$. Let $x_1, x_2 \in V(G)$, with $x_1 \in K_1$, and $x_2 \in K_2$, for some $K_1, K_2 \in \mathcal{K}_G$. We have $D_T(x_1, x_2) = 2 + D_T(\rho_1, \rho_2) = 2 + D_{T_{\mathcal{K}}^G}(K_1, K_2)$. By lemma 3.2 we obtain that $D_T(x_1, x_2) \leq 4 + D_G(x_1, x_2)/c \leq 5D_G(x_1, x_2)$. Also by the same lemma, $D_T(x_1, x_2) \geq D_G(x_1, x_2)/(20c)$. By combining the above it follows that the distortion is at most $100c$.

THEOREM 4.1. *There exists a polynomial time, constant-factor approximation algorithm, for the problem of embedding an unweighted graph into a tree, with minimum multiplicative distortion.*

5 Well-Separated Tree-Like Decompositions

Before we describe our algorithm for embeddings general metrics, we need to introduce a refined decomposition procedure. As in the unweighted case, we want to obtain a partition of the input metric space in a set of clusters, solve the problem independetly for each cluster, and join the solutions to obtain a solution for the input metric.

The key properties of the tree-like decomposition used in the case of unweighted graphs are the following: (1) the distances in the tree of clusters approximate the distances in the original graph, and (2) the diameter of each cluster is small.

Observe that if the graph is weighted with maximum edge weight W_G, and the clusters have small diameter, then the distance between two adjacent clusters of a tree-like decomposition can be any value between 1 and W_G. Thus, the tree of clusters cannot approximate the original distances by a factor better than W_G.

We address this problem by introducing a new decomposition that alows the diameter of each cluster to be arbitrary large, while guaranteeing that (1) the distance between clusters is sufficiently large, and (2) after solving the problem independently for each cluster, the solutions can be merged together to obtain a solution for the input metric.

Formally, let $G = (V, E)$ be a graph that γ-embeds into a tree. Let also $r \in V(G)$, and $\alpha \geq 1$ be a parameter. Intuitively, the parameter α controls the distance between clusters in the resulting partition.

A (r, γ, α)-*well-separated tree-like decomposition* is a triple $(T_{\mathcal{K}}^G, \mathcal{K}_G, \mathcal{A}_G)$, were $(T_{\mathcal{K}}^G, \mathcal{K}_G)$ is a (r, γ)-tree-like decomposition of G, and $\mathcal{A}_G$ is defined as follows.

For a set $A \subseteq V(G)$, let $Z_A = \{K \in \mathcal{K}_G | K \cap A \neq \emptyset\}$. Define $T_{\mathcal{K}}^{G,A}$ to be the vertex-induced subgraph $T_{\mathcal{K}}^G[Z_A]$.

PROPOSITION 5.1. *Let $A \subseteq V(G)$, such that $G[A]$ is connected. Then, $T_{\mathcal{K}}^{G,A}$ is a subtree of $T_{\mathcal{K}}^G$*

Proof. Deferred to the full version of this paper. □

$\mathcal{A}_G$ is computed in two steps:

Step 1. We define a partition $\bar{\mathcal{A}}_G$. $\bar{\mathcal{A}}_G$ contains all the connected components of G obtained after removing all the edges of weight greater than $W_G/(\gamma^{3/2}\alpha)$.

Step 2. We set $\mathcal{A}_G := \bar{\mathcal{A}}_G$. While there exist $A_1, A_2 \in \mathcal{A}_G$ such that the diameter of $T_{\mathcal{K}}^{G,A_1} \cap T_{\mathcal{K}}^{G,A_2}$ is greater than 50γ, we remove A_1, and A_2 from $\mathcal{A}_G$, and we add $A_1 \cup A_2$ in $\mathcal{A}_G$. We repeat until there are no more such pairs A_1, A_2.

5.1 Properties of Well-Separated Tree-Like Decompositions

We now show the main properties of a well-separated tree-like decomposition that will be used by our algorithm for embedding general metrics. They are summarized in the following two lemmata.

Intuitively, the first lemma shows that the distance between different clusters is sufficiently large, and at the same time they don't share long parts of the tree $T_{\mathcal{K}}^G$. The technical importance of the later property will be justified in the next section. It worths mentioning however that intuitively, the fact that the intersections are short will allow us to arrange the clusters of $\mathcal{A}_G$ in a tree, without intersections, incurring only a small distortion.

LEMMA 5.1. *For any $A_1, A_2 \in \mathcal{A}_G$, $D_G(A_1, A_2) \geq W_G/(\gamma^{3/2}\alpha)$, and $T_{\mathcal{K}}^{G,A_1} \cap T_{\mathcal{K}}^{G,A_2}$ is a subtree of $T_{\mathcal{K}}^G$ with diameter at most 50γ.*

Proof. For any $A_1, A_2 \in \bar{\mathcal{A}}_G$, we have that $D(A_1, A_2) \geq W_G/(\gamma^{3/2}\alpha)$. Since $\mathcal{A}_G$ is obtained by only merging sets, the first property holds. Moreover, the construction of $\mathcal{A}_G$ clearly terminates, and the second property follows by the termination condition of the construction procedure. □

The next lemma will be used to argue that when recursing in a cluster, the corresponding induced metric can be sufficiently approximated by a graph with small maximum edge weight.

LEMMA 5.2. *For any $A \in \mathcal{A}_G$, the $W_G/(\gamma^{1/2}\alpha)$-restricted subgraph of $G[A]$, is connected.*

Proof. For an embedding of G into a tree T, and for disjoint $A_1, A_2 \subset V(G)$, we say that A_1 *splits* A_2 *in* T, if A_2 intersects at least 2 connected components of $T[V(G) \setminus A_1]$.

Claim 3. *Let $A_1, A_2 \subset V(G)$, with $A_1 \cap A_2 = \emptyset$, such that $G[A_1]$, and $G[A_2]$ are both connected. Assume that the diameter of $T_{\mathcal{K}}^{G,A_1} \cap T_{\mathcal{K}}^{G,A_2}$ is greater than 50γ. Consider an optimal non-contracting embedding of G into a tree T, with distortion γ. Then, either A_1 splits A_2 in T, or A_2 splits A_1 in T.*

Proof. Since $G[A_1]$, and $G[A_2]$ are both connected, it follows by Proposition 5.1 that $T_{\mathcal{K}}^{G,A_1}$, and $T_{\mathcal{K}}^{G,A_2}$ are both connected subtrees of $T_{\mathcal{K}}^{G}$. Pick a path $p = K_1, K_2, \ldots, K_l$ in $T_{\mathcal{K}}^{G}$, with $l > 50\gamma$, that is contained in $T_{\mathcal{K}}^{G,A_1} \cap T_{\mathcal{K}}^{G,A_2}$.

Assume that the assertion is not true. Let $A_1' = A_1 \cap (\bigcup_{i=1}^{l} K_i)$, and let $A_2' = A_2 \cap (\bigcup_{i=1}^{l} K_i)$. Let T_1 be the minimum connected subtree of T that contains A_1', and similarly let T_2 be the minimum connected subtree of T that contains A_2'. It follows that $T_1 \cap T_2 = \emptyset$.

Let x_1 be the unique vertex of T_1 which is closest to T_2. Since T_1 is minimal, x_1 disconnects T_1. Moreover, since $G[A_1]$ is connected, it follows that there exists $\{w, w'\} \in E(G)$, such that the path from w to w' in T passes through x_1. Since $D_G(w, w') \leq W_G$, we obtain that there exists $x_1^* \in \{w, w'\}$, with $D_T(x_1^*, x_1) \leq D_T(w, w')/2 \leq \gamma D_G(w, w')/2 \leq \gamma W_G/2$.

By Lemma 3.1, it follows that for any $x \in A_1'$, there exists $x' \in A_2'$, such that $D_G(x, x') \leq 20 W_G \gamma$. Moreover, for any $x \in A_1'$, $D_T(x, T_2) = D_T(x, x_1) + D_T(x_1, T_2)$. Thus, for any $x \in A_1'$, $D_T(x, x_1^*) \leq D_T(x_1, x_1^*) + D_T(x, x_1) \leq \gamma W_G/2 + D_T(x, T_2) \leq \gamma W_G/2 + \gamma D_G(x, A_2') \leq 21 W_G \gamma^2$.

Pick $z \in A_1' \cap K_1$, and $z' \in A_1' \cap K_l$. By the triangle inequality, $D_T(z, z') \leq D_T(z, x_1^*) + D_T(x_1^*, z') \leq 42 W_G \gamma^2$. On the other hand, the distance between K_1, and K_l in $T_{\mathcal{K}}^{G}$ is $l-1$. Thus, by Lemma 3.2 we obtain that $D_G(z, z') \geq (l-3) W_G \gamma > 45 W_G \gamma^2$, which contradicts that fact that the embedding of M into T is non-contracting. □

Fix an optimal non-contracting embedding of G into a tree T, with distortion γ.

For $k \geq 0$, let $\mathcal{A}_G^k$ be the partition $\mathcal{A}_G$ after k iterations of Step 2 have been performed, with $\mathcal{A}_G^0 = \bar{\mathcal{A}}_G$.

Assume that the assertion is not true, and pick the smallest k, such that there exists $A \in \mathcal{A}_G^k$, such that the $W_G/(\gamma^{1/2}\alpha)$-restricted subgraph of $G[A]$ is not connected. Assume that A is obtained by joining $A_1, A_2 \in \mathcal{A}_G^{k-1}$. By the minimality of k, it follows that the $W_G/(\gamma^{1/2}\alpha)$-restricted subgraphs of $G[A_1]$, and $G[A_2]$ respectively are connected. Thus, $D_G(A_1, A_2) > W_G/(\gamma^{1/2}\alpha)$.

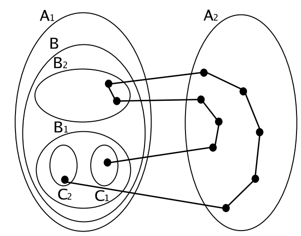

Figure 3: Case 2 of the proof of Lemma 5.2.

By claim 3, we can assume w.l.o.g. that A_2 splits A_1. Thus, by removing A_2 from T, we obtain a collection of connected components F_1. Consider the partition F_1' of A_1 defined by restricting F_1 on A_1. Formally, $F_1' = \{f \cap A_1 | f \in F_1, f \cap A_1 \neq \emptyset\}$. We have to consider the following cases:

Case 1: *There exists $Z \in \bar{\mathcal{A}}_G$, with $Z \subseteq A_1$, such that Z intersects at least two sets in F_1'.* By considering only edges of weight at most $W_G/(\gamma^{3/2}\alpha)$, the induced subgraph $G[Z]$ is connected. It follows that there exist $z_1, z_2 \in Z$, with $D_G(z_1, z_2) \leq W_G/(\gamma^{3/2}\alpha)$, such that the path from z_1 to z_2 in T passes through A_2. Thus, $D_T(z_1, z_2) \geq 2 D_G(A_1, A_2) > 2 W_G/(\gamma^{1/2}\alpha) \geq 2\gamma D(z_1, z_2)$, contradicting the fact that the expansion of T is at most γ.

Case 2: *For any $Z \in \bar{\mathcal{A}}_G$, with $Z \subseteq A_1$, we have $Z \subseteq Z'$, for some $Z' \in F_1'$.* Observe that for ant $t \geq 0$, any element in $\mathcal{A}_G^t$ is obtained as the union of elements of $\bar{\mathcal{A}}_G$. Thus, we can pick the minimum $j \geq 1$, such that there exist $B_1, B_2 \in \mathcal{A}_G^{j-1}$, such that during iteration j of Step 2, the set $B = B_1 \cup B_2$ is obtained, with $B \subseteq A_1$, and such that $B_1 \subseteq Z_1'$, and $B_2 \subseteq Z_2'$, for some $Z_1', Z_2' \in F_1'$, In other words, we pick the minimum j such that we can find sets $B_1, B_2 \in \mathcal{A}_G^{j-1}$, that are contained in A_2, and neither of them is split by A_2 in T. W.l.o.g., we can assume that B_2 splits B_1 in T. Thus, there exist $C_1, C_2 \subseteq B_1$, such that any path between C_1 and C_2 in T passes through B_2. Moreover, any path from B_1 to B_2 in T passes through A_2. Thus, any path from C_1 to C_2 in T passes through A_2. This however contradicts the minimality of j. The scenario is depicted in Fig 3. □

6 Approximation Algorithm for Embedding General Metrics

In this section we present an approximation algorithm for embedding general metrics into trees. Before we get into the technical details of the algorithm, we give an informal description. The main idea is to partition the input metric M using a well-separated tree-like decomposition, and then solve the problem independently for each cluster of the partition by recursion. After solving all the sub-problems, we can combine the partial solutions to obtain a solution for M. There are a few points that need to be highlighted:

Termination of the recursion. As pointed out in the description of the well-separated tree-like decompositions, the clusters of the resulting partition might have arbitrarily long diameter. In particular, we cannot guarantee that by recursively decomposing each cluster we obtain sub-clusters of smaller diameter. To that extend, our recursion deviates from standard techniques since the sub-problems are not necessarily smaller in a usual sense. Instead, our decomposition procedure guarantees that at each recursive step, the metric of each cluster can be approximated by a graph with smaller maximum edge length. This can be thought as restricting the problem to a smaller metric scale.

Merging the partial solutions. The partial solution for each cluster in the recursion is an embedding of the cluster into a tree. As in the algorithm for unweighted graphs, we merge the partial solutions using the tree $T_{\mathcal{K}}^{G}$ of the well-separated tree-like decomposition as a rough approximation of the resulting tree. However, in the case of a well-separated decomposition, the parts of $T_{\mathcal{K}}^{G}$ that correspond to different clusters of the partition $\mathcal{A}_G$ might overlap. Moreover, since some of the clusters might be long, we need to develop an elaborate procedure for merging the different trees into a tree for M, without incurring large distortion.

6.1 The Main Inductive Step We will now describe the main inductive step of the algorithm. Let $M = (X, D)$ be a finite metric that c-embeds into a tree. At each recursive step performed on a cluster A^* of M, the algorithm is given a graph G with vertex set A, that c-approximates M. In order to recurse in sub-problems, we compute a well-separated tree-like decomposition of G. We chose the parameters of the well-separated decomposition so that each sub-cluster A, can be c-approximated by a graph that has maximum edge weight significantly smaller than the maximum edge weight of G. Formally, the main recursive step is as follows.

Procedure RecursiveTree

Input: A graph G with maximum edge weight W_G, that c-approximates M.

Output: An embedding of G into a tree S.

Step 1: Partitioning. If G contains only one vertex, then we output a trivial tree containing only this vertex. Otherwise, we proceed as follows. We pick $r \in V(G)$, and compute a (r, c^2, α)-well-separated tree-like decomposition $(T_{\mathcal{K}}^{G}, \mathcal{K}_G, \mathcal{A}_G)$ of G, where $\alpha > 0$ will be determined later.

Step 2: Recursion. For any $A \in \mathcal{A}_G$, let G_A be the W_G/α-restricted subgraph, with $V(G_A) = A$. We recursively execute the procedure RecursiveTree on G_A, and we obtain a tree S^A.

Step 3: Merging the solutions. In this final step we merge the trees S^A to obtain S.

We define a tree T as follows. We first remove from $T_{\mathcal{K}}^{G}$ all the edges between vertices at level $i50c^2$, and $i50c^2+1$, for any integer $i : 1 \leq i \leq n/(50c^2)$. For any connected component C of the resulting forest, T contains a vertex C. Two vertices $C, C' \in V(T)$ are connected, iff there is an edge between C, and C' is $T_{\mathcal{K}}^{G}$. We consider T to be rooted at the vertex which corresponds to the subtree of $T_{\mathcal{K}}^{G}$ that contains r. Furthermore, for each $A_i \in \mathcal{A}_G$, we define a subtree T_i of T as follows: T_i contains all the vertices C of T, such that $T_{\mathcal{K}}^{G,A_i}$ visits C.

Lemma 6.1. *There exists a polynomial-time algorithm that computes an unweighted tree T', and for any $i \in [k]$ a mapping $\phi_i : V(T_i) \to V(T')$, such that*

- *for any $i, j \in [k]$, $\phi_i(T_i) \cap \phi_j(T_j) = \emptyset$,*
- *for any $i, j \in [k]$, for any $v_i \in V(T_i)$, and $v_j \in V(T_j)$, $D_T(v_i, v_j) \leq D_{T'}(\phi_i(v_i), \phi_j(v_j)) \leq 20(D_T(v_i, v_j) + 1) \log n$.*

Proof. Deferred to the full version of this paper. □

Note that the tree T' might contain vertices $C \in V(T)$, such that for any $K \in \mathcal{K}_G$, $K \notin C$. We call such a vertex *steiner*. First, for each steiner vertex $C \in V(T')$ we add a vertex $v_C \in V(S)$. We have to add the following types of edges:

- For any $C_1, C_2 \in V(T')$, such that both C_1, and C_2 are steiner vertices, we add the edge $\{v_{C_1}, v_{C_2}\}$ in S, with weight $W_G/(c^3\alpha)$.
- For any $C_1, C_2 \in V(T')$, such that C_2, is a steiner vertex, and there exists $A_1 \in \mathcal{A}_G$, such that $C_1 \in \phi_1(T_1)$, we pick $K_1 \in T_{\mathcal{K}}^{G,A_1}$, with $K_1 \in C_1$, and an arbitrary $x_1 \in K_1$, and we add the edge $\{x_1, v_{C_1}\}$ in S. The weight of this new edge is $W_G/(c^3\alpha)$.

- For any pair $A_1, A_2 \in \mathcal{A}_G$, with $A_1 \neq A_2$, such that there exists an edge in T' connecting $\phi_1(T_1)$ with $\phi_2(T_2)$, we add an edge between S^{A_1}, and S^{A_2}. We pick the edge that connects S^{A_1} with S^{A_2} as follows. Pick $C_1, C_2 \in V(T)$, with $C_1 \in T_1$, and $C_2 \in T_2$, such that there is an edge between $\phi_1(C_1)$, and $\phi_2(C_2)$ in T'. We pick an arbitrary pair of points x_1, x_2, with $x_1 \in K_1 \in C_1$, and $x_2 \in K_2 \in C_2$, for some $K_1, K_2 \in \mathcal{K}_G$, and we connect S^{A_1} with S^{A_2} by adding the edge $\{x_1, x_2\}$ of length $D(x_1, x_2)$.

Given the metric $M = (X, D)$, the algorithm first computes a weighted complete graph $G_0 = (V, E)$, with $V(G_0) = X$, such that the weight of each edge $\{u, v\} \in E(G)$ is equal to $D(u, v)$. Let Δ be the diameter of M. Clearly, G_0 is a Δ-restricted subgraph. The algorithm then executes the procedure RecursiveTree on G_0, and outputs the resulting tree S.

Before we bound the distortion of the resulting embedding, we first need to show that at each recursive call of the procedure RecursiveTree, the graph G satisfies the input requirements. Namely, we have to show that G c-approximates M. Clearly, this holds for G_0. Thus, it suffices to show that the property is maintained for each graph G_A, were $A \in \mathcal{A}_G$. Observe that since G c-approximates M, and M c-embeds into a tree, it follows that G c^2-embeds into a tree. Since $(T^G_{\mathcal{K}}, \mathcal{K}_G, \mathcal{A}_G)$ is a (r, c^2)-well-separated decomposition, we can assume the properties of lemmata 5.1, and 5.2, for $\gamma = c^2$.

Lemma 6.2. *For any $A \in \mathcal{A}_G$, G_A c-approximates M.*

Proof. Deferred to the full version of this paper. □

The next two lemmata bound the distortion of the resulting embedding of G into S. The fact that the contraction is small follows by the fact that the distance between the clusters in $\mathcal{A}_G$ is sufficiently large. The expansion on the other hand, depends on the maximum depth of the recursion. This is because at each recursive call, when we merge the trees S^A to obtain S, we incurr an extra $c^{O(1)} \log n$-factor in the distortion. Since at every recursive call the maximum edge weight of the input graph decreases by a factor of α, the parameter α can be used to adjust the recursion depth in order to optimize the final distortion.

Lemma 6.3. *The contraction of S is $O(c^7 \alpha)$.*

Proof. Deferred to the full version of this paper. □

Lemma 6.4. *The expansion of S is at most $(c^{O(1)} \log n)^{\log_\alpha \Delta}$.*

Proof. Deferred to the full version of this paper. □

Theorem 6.1. *There exists a polynomial-time algorithm which given a metric $M = (X, D)$ that c-embeds into a tree, computes an embedding of M into a tree, with distortion $(c \log n)^{O(\sqrt{\log \Delta})}$.*

Proof. By Lemmata 6.3, and 6.4, it follows that the distortion of S is $c^{O(1)} \alpha (c^{O(1)} \log n)^{\log_\alpha \Delta}$. By setting $\alpha = 2^{\sqrt{\log \Delta}}$, we obtain that the distortion is at most $(c \log n)^{O(\sqrt{\log \Delta})}$. □

7 Composing Relative Embeddings

In this section we are going to obtain a polynomial time algorithm for embedding a metric M into the line. The idea of the algorithm is to embed the metric first into a tree metric using the previous algorithm and then use [BCIS05] to embed the tree into the line. The approximation factor that we get is going to be better than the best known result [BCIS05].

Let F, F' be families of n-point metric spaces. We say that an algorithm A is an *$\alpha(c)$-distortion algorithm from F to F'*, if on input $X \in F$, it outputs $X' \in F'$, and an embedding $f : X \to X'$, with distortion $\alpha(c)$, where c is the optimal distortion for embedding X into a metric in F'. We also say that *F β-embeds into F'*, if for any $X \in F$, there exists $X' \in F'$, such that X can be embedded into X', with distortion at most β.

Lemma 7.1. *Let F_1, F_2, F_3 be families of n-point metric spaces, such that F_3 β-embeds into F_2. Let A_1 be an $\alpha_1(c)$-distortion algorithm from F_1 to F_2, and let A_2 be an $\alpha_2(c)$-distortion algorithm from F_2 to F_3. Then, there exists a $\beta \cdot c \cdot \alpha_2(c \cdot \alpha_1(\beta \cdot c))$-algorithm from F_1 to F_3.*

Proof. Assume that we are given X_1 that c-embeds into F_3. It follows that X_1 embeds into F_2 with distortion $\beta \cdot c$. We compute using A_1 an embedding f_1 of X_1 into $X_2 \in F_2$, with distortion $\alpha_1(\beta \cdot c)$. In other words, the distances in X_2 $\alpha_1(\beta c)$-approximate the distances in X_1. Therefore, X_2 embeds into F_3 with distortion at most $d = c \cdot \alpha_1(\beta \cdot c)$. Using A_2, we compute an embedding f_2 of X_2 into $X_3 \in F_3$, with distortion $\alpha_2(d) = \alpha_2(c \cdot \alpha_1(\beta \cdot c))$. Since X_2 $\alpha_1(\beta c)$-approximates X_1, it follows that the composition $f_2 \circ f_1$ is an embedding of X_1 into F_3, with distortion at most $\beta \cdot c \cdot \alpha_2(c \cdot \alpha_1(\beta \cdot c))$. □

Corollary 7.1. *There exists a polynomial-time algorithm that given a metric M of spread Δ that c-embeds into the line, computes an embedding of M into the line with distortion $(c \log n)^{O(\sqrt{\log \Delta})}$.*

Proof. We apply Lemma 7.1 with F_1 the family of all n-point metrics of spread at most Δ, F_2 the family of all n-point trees, and F_3 the family of all n-point line metrics. A_1 is the algorithm given in Theorem 6.1, A_2 is the $c^{O(1)}$-distortion algorithm for embedding trees into the line given in [BCIS05], and $\beta = 1$, since each line metric is also a tree metric. □

Acknowledgments We thank Julia Chuzhoy for many insightful discussions about the problem.

References

[ABD+05] N. Alon, M. Bădoiu, E. Demaine, M. Farach-Colton, M. T. Hajiaghayi, and A. Sidiropoulos. Ordinal embeddings of minimum relaxation: General properties, trees and ultrametrics. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2005.

[ABFC+96] R. Agarwala, V. Bafna, M. Farach-Colton, B. Narayanan, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy: (fitting distances by tree metrics). *7th Symposium on Discrete Algorithms*, 1996.

[AC05] Nir Ailon and Moses Charikar. Fitting tree metrics: Hierarchical clustering and phylogeny. In *Annual Symposium on Foundations of Computer Science*, 2005.

[BCIS05] M. Bădoiu, J. Chuzhoy, P. Indyk, and A. Sidiropoulos. Low-distortion embeddings of general metrics into the line. *Annual ACM Symposium on Theory of Computing*, 2005.

[BCIS06] Mihai Badoiu, Julia Chuzhoy, Piotr Indyk, and Anastasios Sidiropoulos. Embedding ultrametrics into low-dimensional spaces. In *Proceedings of the ACM Symposium on Computational Geometry*, 2006.

[BDG+05] M. Bădoiu, K. Dhamdhere, A. Gupta, Y. Rabinovich, H. Raecke, R. Ravi, and A. Sidiropoulos. Approximation algorithms for low-distortion embeddings into low-dimensional spaces. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2005.

[CC95] Leizhen Cai and Derek G. Corneil. Tree spanners. *SIAM J. Discrete Math*, 8(3):359–387, 1995.

[DEKM98] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis*. Cambridge University Press, 1998.

[EP04] Y. Emek and D. Peleg. Approximating minimum max-stretch spanning trees on unweighted graphs. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2004.

[Epp00] D. Eppstein. Spanning trees and spanners. *Handbook of Computational Geometry (Ed. J.-R. Sack and J. Urrutia)*, pages 425–461, 2000.

[FCKW93] M. Farach-Colton, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary tree. *Annual ACM Symposium on Theory of Computing*, 1993.

[FK01] S. P. Fekete and J. Kremer. Tree spanners in planar graphs. *Discrete Applied Mathematics*, 108:85–103, 2001.

[Gup01] A. Gupta. Steiner nodes in trees don't (really) help. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[HKM05] Boulos Harb, Sampath Kannan, and Andrew McGregor. Approximating the best-fit tree under l_p norms. In *Proceedings of Approx*, 2005.

[HP05] Alexander Hall and Christos H. Papadimitriou. Approximating the distortion. In *APPROX-RANDOM*, pages 111–122, 2005.

[Ind01] P. Indyk. Tutorial: Algorithmic applications of low-distortion geometric embeddings. *Annual Symposium on Foundations of Computer Science*, 2001.

[KRS04] C. Kenyon, Y. Rabani, and A. Sinclair. Low distortion maps between point sets. *Annual ACM Symposium on Theory of Computing*, 2004.

[KTT98] A. Kearsley, R. Tapia, and M. Trosset. The solution of the metric stress and sstress problems in multidimensional scaling using newtons method. *Computational Statistics*, 1998.

[Lin02] N. Linial. Finite metric spaces - combinatorics, geometry and algorithms. *Proc. International Congress of Mathematicians III*, pages 573–586, 2002.

[LLR94] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Proc. 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 577–591, 1994.

[LNP06] J. R. Lee, Assaf Naor, and Y. Peres. Trees and markov convexity. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[Mat90] J. Matoušek. Bi-lipschitz embeddings into low-dimensional euclidean spaces. *Comment. Math. Univ. Carolinae*, 31:589–600, 1990.

[MDS] MDS: Working Group on Algorithms for Multidimensional Scaling. Algorithms for multidimensional scaling. DIMACS Web Page.

[PR98] D. Peleg and E. Reshef. A variant of the arrow distributed directory protocol with low average case complexity. In *Proc. 25th Int. Colloq. on Automata, Language and Programming*, pages 670–681, 1998.

[PS05] C. Papadimitriou and S. Safra. The complexity of low-distortion embeddings between point sets. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 112–118, 2005.

[PT01] D. Peleg and D. Tendler. Low stretch spanning trees for planar graphs. *Technical Report MCS01-14, The Weizmann Institute of Science*, 2001.

[PU87] D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. In *Proc. 6th ACM Symposium on Principles of Distributed Computing*, pages 77–85, 1987.

[Sci05] Will there ever be a tree of life that systematists can agree on? *Science, 125th Anniversary Issue. Available at* `http://www.sciencemag.org/sciext/125th/`, 2005.

[VRM+97] G. Venkatesan, U. Rotics, M. S. Madanlal, J. A. Makowsky, and C. P. Rangan. Restrictions of minimum spanner problems. *Information and Computation*, 136(2):143–164, 1997.