Reconstructing Approximate Tree Metrics*

Ittai Abraham
The Hebrew University of Jerusalem
Israel
ittaia@cs.huji.ac.il

Mahesh Balakrishnan
Cornell University
Ithaca, NY
mahesh@cs.cornell.edu

Fabian Kuhn
Institute of Theoretical Computer Science
ETH Zurich, Switzerland
kuhn@inf.ethz.ch

Dahlia Malkhi
Microsoft Research Silicon Valley
Mountain View, CA
dalia@microsoft.com

Venugopalan Ramasubramanian
Microsoft Research Silicon Valley
Mountain View, CA
rama@microsoft.com

Kunal Talwar
Microsoft Research Silicon Valley
Mountain View, CA
kunal@microsoft.com

ABSTRACT

We introduce a novel measure called *ε-four-points condition* (ε-4PC), which assigns a value $\varepsilon \in [0, 1]$ to every metric space quantifying how close the metric is to a tree metric. Data-sets taken from real Internet measurements indicate remarkable closeness of Internet latencies to tree metrics based on this condition. We study embeddings of ε-4PC metric spaces into trees and prove tight upper and lower bounds. Specifically, we show that there are constants c_1 and c_2 such that, (1) every metric (X, d) which satisfies the ε-4PC can be embedded into a tree with distortion $(1 + \varepsilon)^{c_1 \log |X|}$, and (2) for every $\varepsilon \in [0, 1]$ and any number of nodes, there is a metric space (X, d) satisfying the ε-4PC that does not embed into a tree with distortion less than $(1 + \varepsilon)^{c_2 \log |X|}$. In addition, we prove a lower bound on approximate distance labelings of ε-4PC metrics, and give tight bounds for tree embeddings with additive error guarantees.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*computations on discrete structures*;
G.2.2 [**Discrete Mathematics**]: Graph Theory—*graph algorithms*;
G.2.2 [**Discrete Mathematics**]: Graph Theory—*network problems*

General Terms

Algorithms, Theory

Keywords

Metric Spaces, Embeddings, Tree Metrics, Four-Points Condition

*Part of this work was done when the first three authors were at MSR Silicon Valley.

PODC'07, August 12–15, 2007, Portland, Oregon, USA.

1. INTRODUCTION

Several distributed systems (e.g., peer-to-peer overlays) and Internet applications (e.g., content distribution networks) benefit from the ability to estimate network latency between end hosts instantaneously, without incurring the overhead of recurrent measurements. Current solutions for latency prediction typically involve embedding the network into a low-dimensional coordinate space (such as Euclidean) through a small number of latency measurements to a subset of the end hosts [8, 9, 30, 19, 23, 25, 29].

In this paper, we study a more intuitive and practical alternative for predicting network latency, namely *embedding into a tree metric*. A metric V is a *tree metric* if there exists a tree with non-negative weights such that $V \subseteq T$ and $d_V(u, v) = d_T(u, v)$ for all $u, v \in V$. Note that the embedded tree might contain additional Steiner nodes not in V. A tree embedding is more intuitive because even though the Internet is not exactly a tree, it has an inherent hierarchy in the relationships between end hosts and Internet Service Providers (ISPs) at different tiers (Tier 1, Tier 2, and Tier 3). It is more practical because trees have been proposed as a basic primitive in distributed systems for multi-cast communication, locality-aware clustering, and data aggregation. Moreover, trees provide the same ability as coordinate-based systems for instantaneous distance estimation through short and efficient distance labels [13, 24].

The primary contribution of this paper is a novel measure called the *ε-four-points condition* (ε-4PC) that quantifies how close a network is to a tree. More precisely, ε-4PC assigns a value $\varepsilon \in [0, 1]$ to every metric space, where $\varepsilon = 0$ indicates that the metric space is an exact tree metric. We present evidence from analysis of real Internet latency measurements that Internet latencies closely approximate a tree metric even though the Internet does not exactly embed into a tree.

The rest of the paper studies the accuracy or *distortion* of embedding network distances into tree metrics within the context of ε-4PC. In particular, we prove asymptomatically tight upper and lower bounds of $(1+\varepsilon)^{\Theta(\log |V|)}$ on the distortion of embedding V into a tree metric, for all metrics $V \in \varepsilon$-4PC. Our proof for the upper bound is constructive, and provides an algorithm for generating a tree embedding.

1.1 Definitions and Summary of Contributions

An *embedding* of a metric space U into another metric space V is a mapping between the two metric spaces $f : U \rightarrow V$. The *distortion* of an embedding f is the worst case ratio of the distance between two nodes in the original space and the corresponding distance in the target space, that is,

$$dist(f) = \frac{\max_{u,v \in \binom{V}{2}} d_V(f(u), f(v))/d_U(u,v)}{\min_{u,v \in \binom{V}{2}} d_V(f(u), f(v))/d_U(u,v)}.$$

A metric space is a *tree metric* if and only if the sub-metric induced by any four points is a tree metric [7]. This observation is generally known as the *Four-Points Condition* (4PC). It states that for any four points, out of the three possible matchings, the two matchings that have the maximum weight have equal weight:

DEFINITION 1.1 (**4-Points Condition (4PC) [7]**). *A metric space (V, d) satisfies the 4PC if for every four points $w, x, y, z \in V$ that are ordered such that $d(w, x)+d(y, z) \leq d(w, y)+d(x, z) \leq d(w, z) + d(x, y)$, $d(w, y) + d(x, z) = d(w, z) + d(x, y)$ holds.*

We propose the following relaxation of the 4PC, where the maximum weight matching is not very far from the second maximum weight matching:

DEFINITION 1.2 (**ε-4-Points Condition (ε-4PC)**). *A metric space (V, d) satisfies the ε-4PC for an arbitrary parameter $\varepsilon \in [0, 1]$ if for every four points $w, x, y, z \in V$ that are ordered such that $d(w, x) + d(y, z) \leq d(w, y) + d(x, z) \leq d(w, z) + d(x, y)$, the following holds:*

$$d(w, z)+d(x, y) \leq d(w, y)+d(x, z)+2\varepsilon\cdot\min\{d(w, x), d(y, z)\}.$$

The parameter ε quantifies the closeness of a metric to a tree metric. On the one hand, for $\varepsilon = 0$, the ε-4PC is exactly the 4PC and thus a metric space with $\varepsilon = 0$ is a tree metric. On the other hand, every metric space satisfies the ε-4PC for $\varepsilon = 1$ since the ε-4PC for $\varepsilon = 1$ follows from the triangle inequality. Note that the ε-4PC, just like the 4PC, is easy to verify as the condition only depends on end-to-end distances between arbitrary node pairs, which can be directly measured in the Internet.

The above definition of a relaxed tree metric is closely related to a prior relaxation called δ-hyperbolicity proposed by Gromov [14].

DEFINITION 1.3 (**δ-hyperbolicity [14]**). *A metric space (V, d) is δ-hyperbolic for a constant $\delta > 0$ if for every four points $w, x, y, z \in V$ that are ordered such that $d(w, x) + d(y, z) \leq d(w, y) + d(x, z) \leq d(w, z) + d(x, y)$, $d(w, z) + d(x, y) \leq d(w, y) + d(x, z) + \delta$ holds.*

Note that in contrast to the hyperbolicity condition, the ε-four-points condition is invariant to scaling. As a consequence, the ε-4PC is a condition about the embeddability into tree metrics with respect to multiplicative distortion whereas hyperbolicity is a condition about the embeddability into tree metrics with respect to additive distortion. Indeed if a metric space (V, d) satisfies the ε-4PC, it can be shown that every four-point submetric of (V, d) has a tree embedding with distortion at most $1 + 2\varepsilon$. However, the ε-4PC is stronger than assuming that every four points of a metric embed into a tree with distortion $1 + c\varepsilon$ for some constant c. As an example, consider the shortest path metric (V, d) of a 4-cycle with edge lengths $\lambda \ll 1$, 1, 1, and 1. The metric $((V, d)$ has a tree embedding with distortion $1 + \lambda$, however the ε-4PC is only satisfied for $\varepsilon = 1$. The δ-hyperbolicity condition is equivalent to assuming that every 4 points embed into a tree metric with additive distortion $O(\delta)$.

In practice, however, the ε-4PC is more useful than the δ-hyperbolicity condition as a measure for closeness to a tree-metric. Unlike the latter, the ε is a bounded parameter whose closeness to 0 has a strong correlation with how close a metric is to a tree-metric. In the latter, even metrics with small δ might be very far from a tree-metric.

We derive the following key upper and lower bounds for embedding metrics with ε-4PC into tree metrics (note that unless explicitly stated, all logarithms are to base 2 throughout the paper):

THEOREM 1.1 (**Upper bound**). *There is a constant c_1 such that any n point metric space that satisfies the ε-four-points condition for a given $\varepsilon \in [0, 1]$ can be embedded into a tree metric with a maximum distortion of $(1 + \varepsilon)^{c_1 \cdot \log n}$.*

THEOREM 1.2 (**Lower bound**). *There is a constant c_2 such that for any $\delta \in [0, 1]$ and any $n > 1$ there exists a metric space $\mathcal{X}_\delta$ on n points that satisfies the ε-four-points condition for $\varepsilon = (e-1)/\ln 2 \cdot \delta$ such that every embedding of $\mathcal{X}_\delta$ into a tree metric has distortion larger than $(1 + \varepsilon)^{c_2 \cdot \log n}$.*

Theorems 1.1 and 1.2 are proven in Sections 2 and 3, respectively. In Section 3, we also show how to extend a distance labeling lower bound for hyperbolic metric spaces (cf. Definition 1.3) from [12] to ε-4PC metrics. We prove that every $(1+\varepsilon)^{\log k}$-approximate distance labeling requires labels of at least $n^{\Omega(1/k)}$ bits.

In addition, we extend results from [14] and [2] for tree embeddings with additive stretch in Section 4. We show that the family of metrics for which every k-point submetric embeds into a tree metric with additive distortion δ embeds into tree metrics with additive distortion $\Theta(\delta \log_k n)$.

1.2 Treeness of the Internet

Before we derive the upper and lower bounds, we first present evidence to show that tree embeddings are relevant for latency prediction in the Internet. Indirectly, some experimental evidence for this has been provided in [28] and [6] where it is argued that embeddings of Internet latencies into hyperbolic geometries achieve better accuracy than embeddings into Euclidean geometries.

Figure 1.2 shows the cumulative distribution (CDF) of ε values for latencies between random sets of four servers on PlanetLab [5], a planetary-scale distributed platform; the latencies are based on the minimum latency over a two month segment, spanning April and May 2006, of the University of Cincinnati all-pairs latency dataset [32]. For comparison, we also plot the CDF of ε values for latencies between random nodes drawn from a spherical coordinate space, which models a network whose latencies are proportional to distances on the surface of a sphere.

Figure 1.2 shows that PlanetLab latencies are closer to a tree metric a large number of quadruplets (75 %) have small ε (< 0.3) with up to 45 % of quadruplets having ε less than 0.1. Compared to PlanetLab, distances on the Sphere typically have significantly higher values of ε. Note that in Figure 1.2, about 5 % of quadruplets have ε greater than 1, which indicates that triangle inequality is not always preserved in the Internet. Despite occasional triangle inequality violations, Figure 1.2 indicates that the Internet is rather 'tree-like.'

1.3 Related Work

For general metrics, it is known that the cycle requires linear distortion for embedding into a tree [26, 15]. To overcome this fact, embedding into a *distribution* of tree metrics (in fact of ultrametrics) has been studied (see [4, 11, 10]). These results imply the existence of an embedding from a weighted graph into a tree, whose average distortion over all edges is logarithmic. Recently [1] give an embedding from a weighted graph into one of its spanning trees, whose average distortion over all pairs is constant. Nevertheless,

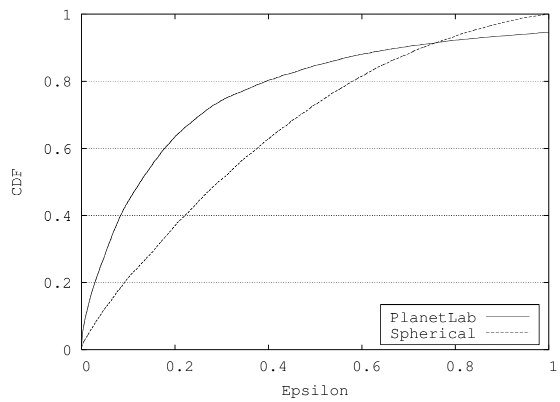

Figure 1: CDF of ε for nodes on PlanetLab vs. nodes on a Sphere: Latencies on PlanetLab are much closer to a tree metric than distances on the surface of a sphere indicating that the Internet is rather 'tree-like'.

all these results apply for arbitrary metrics and hence are bounded by the worst case linear distortion lower bounds.

Our result can be cast in a more general "local versus global" question of metric space embeddings: Given that a property is held "locally" for any set of 4 points what can be said about the "global" structure of the metric? In [3], a comprehensive study of this theme is conducted. One of the author's results is that if any set of k points $1+\varepsilon$ embed into a ultra-metric then the whole n point space $(1+\varepsilon)^{O(\log_k n)}$ embeds into an ultra-metric. Obtaining similar results if any k points $1+\varepsilon$-embed into a tree-metric is left open. In this context, our results provide a partial answer since every 4 points of an ε-4PC metric $(1+2\varepsilon)$-embed into a tree metric.

The study of δ-hyperbolic metric spaces and their (coarse) embedding was initiated by Gromov in [14]. Recently, distance labeling [12] as well as a variety of algorithms [18] for δ-hyperbolic metric spaces have been studied.

Our upper bound results provide a bound on the distortion for the family of all ε-4PC metrics for a given ε. A related question is that of algorithmically finding the best embedding of a fixed metric in polynomial time. This question is studied by [2], given a metric whose best embedding into a tree metric obtains additive distortion ϵ the authors prove that it is NP hard to obtain a $9\epsilon/8$ additive distortion and give a polynomial time embedding with 3ϵ additive distortion.

Kleinberg, Slivkins, and Wexler [17] suggest an alternative approach to explain why internet latencies can be efficiently embedded. They study new notions of embedding with slack. In [31], a system inspired by work on low-dimensional metrics aims to obtain efficient latency prediction.

In Bioinformatics, reconstructing tree metrics using a technique similar to that of Buneman [7] is often called *neighbor-joining*. This technique is applied for the creation of phylogenetic trees based on DNA or protein sequence data (see [27, 22]).

Generally, there has been comprehensive research on low distortion embeddings (see [20, 21]). Typically the goal is to provide low distortion embedding of a "complex" metric space into a "simpler" metric space. Such embeddings have found numerous applications in approximation algorithms (see [16] for a survey).

Algorithm 1 Basic Tree Construction

```
1: select arbitrary root node r ∈ V;
2: return tree T := constructTree(V \ {r}, r);
3:
4: function constructTree(V,r):
5:   if |V| > 1 then
6:     choose p, q ∈ V s.t. (p|q)_r = max_{u,v∈V} {(u|v)_r}
7:     T := constructTree(V \ {q}, r);
8:     add Steiner node t_q at distance (q|r)_p from p;
9:     add node q at distance (p|r)_q from t_q
10:  else
11:    T is line segment from v ∈ V to r;
12:  fi;
13:  return T
14: end constructTree
```

2. ALGORITHM

In this section, we describe an algorithm that, given a metric space (V, d) satisfying the ε-4PC for some $\varepsilon \leq 1$, constructs a Steiner tree T on which the distances of (V, d) are distorted by a factor of at most $(1+\varepsilon)^{O(\log n)}$ where $n = |V|$. Our algorithm can be seen as a variant of the algorithm described by Buneman for computing the Steiner tree representing a given tree metric [7].

First, observe that the distances among every set of three points x, y, and z of a metric space can be exactly represented by a tree by adding one additional Steiner node t. The three nodes x, y, and z are all connected to t. The distance from x to t is $(d(x,y) + d(x,z) - d(y,z))/2$ and the distances from y and z to t are set similarly. It can be verified that x, y, and z have the correct distances to each other. Note that the distances between t and the three nodes are non-negative because of the triangle inequality. For simplicity, we introduce a notation which was introduced by Gromov in his work on hyperbolic metric spaces [14].

DEFINITION 2.1 (**Gromov Product**). *Let $x, y, z \in X$ be three points of a metric space (X, d). The Gromov product $(x|y)_z$ is*

$$(x|y)_z = \frac{1}{2} \cdot \big(d(x,z) + d(y,z) - d(x,y)\big).$$

Hence, the Gromov product $(x|y)_z$ is the distance of z to the Steiner node t on the tree connecting x, y, and z described above. The basic idea of Buneman's consruction (and of our algorithm) is to compute the Steiner nodes from the leaves towards the center of the tree. One way to do this, is to fix some node r as root and to compute the Steiner nodes in decreasing distance to r. The distance of the Steiner node connecting r to the path between two nodes p and q is $(p|q)_r$. For the first Steiner node t, we therefore have to find p and q such that $(p|q)_r$ is maximized. The nodes p and q then are two leaves that are directly attached to t. This leads to the basic tree construction given by Algorithm 1. It selects p and q such that $(p|q)_r$ is maximized, removes q from the metric, constructs the tree for the resulting $(n-1)$-point metric, and adds the removed node to the recursively constructed tree by adding the Steiner node t such that the distances from t to p, q, and r are $(q|r)_p$, $(p|r)_q$, and $(p|q)_r$ respectively.

We can make the following basic observations about Algorithm 1. All nodes are added to the tree such that they have the correct distance to the root node r, i.e., $T(u,r) = d(u,r)$ for all $u \in V$. In addition, if p and q are nodes chosen in Line 6 of some execution of **constructTree**(X, r), we also have $T(p,q) = d(p,q)$. Both statements follow by induction and the way q is added to the tree in Line 9. By the maximality condition in Line 6 of Algorithm 1, Steiner nodes are computed in decreasing distance to r, and are therefore inserted in increasing distance to r. Hence, when Steiner node t_q is added to the tree in Line 11, there is no Steiner node between t_q and p. For a node x that is already in T when node q is added, we therefore have

$$T(q,x) = T(p,x) - (q|r)_p + (p|r)_q = T(p,x) - d(p,r) + d(q,r). \tag{1}$$

Equation (1) together with the ε-4PC applied to p, q, r, and x will be the key to analyze the quality of the tree constructed by Algorithm 1 as shown by Lemma 2.1 below.

For the analysis of the algorithm, let us now assume that we are given a metric space (V,d) that satisfies the ε-4PC for some $\varepsilon \in [0,1]$. Let $T(u,v)$ be the distance between u and v on the constructed tree and let $\delta(u,v) := T(u,v) - d(u,v)$ be the additive error of the distance between u and v in the tree. Further, we define a function $\alpha : V \to V$ which characterizes the dependencies between nodes in the tree construction. Let p and q be two nodes chosen in Line 6 of an execution of **constructTree**(V,r). Node q is added to the tree with respect to the position of p in the tree. We define $\alpha(q) = p$. For the last two nodes v and r that are added in Line 11, we set $\alpha(v) = \alpha(r) = r$. The next lemma bounds the error from a single execution of **constructTree**(V,r).

LEMMA 2.1. *Let $x, y \in V$ be two arbitrary nodes such that in the construction of the tree T, x is removed from the set V before y is removed from V. We can bound the error $\delta(x,y)$ as follows:*

$$\delta(\alpha(x),y) - 2\varepsilon d(x,\alpha(x)) \le \delta(x,y) \le \delta(\alpha(x),y) + 2\varepsilon d(x,\alpha(x)).$$

PROOF. Let us look at a call of **constructTree**(V,r) where x is removed from V. The nodes p and q in **construcTree**(V,r) are therefore x and $\alpha(x)$. By the maximality condition in Line 6,

$$\begin{aligned} d(x,\alpha(x)) + d(r,y) &\le d(x,r) + d(\alpha(x),y), \\ d(x,\alpha(x)) + d(r,y) &\le d(x,y) + d(\alpha(x),r). \end{aligned}$$

Applying ε-four-point condition, we obtain

$$\begin{aligned} d(\alpha(x),y) - 2\varepsilon d(x,\alpha(x)) &\le d(x,y) + d(\alpha(x),r) - d(x,r) \\ &\le d(\alpha(x),y) + 2\varepsilon d(x,\alpha(x)). \end{aligned} \tag{2}$$

For the tree distances, we get

$$\begin{aligned} T(\alpha(x),y) &= T(x,y) + T(\alpha(x),r) - T(x,r) \\ &= T(x,y) + d(\alpha(x),r) - d(x,r). \end{aligned} \tag{3}$$

Remember that the tree T is constructed such that tree distances to r are correct, that is $T(u,r) = d(u,r)$ for all $u \in V$. Combining Inequality (2) and Equation (3) now implies the claim. □

Let $\mathcal{T}$ be the tree rooted at r which is defined by $\alpha(v)$ such that u is the parent node of v if $u = \alpha(v)$. Let z be the least common ancestor of two nodes x and y in $\mathcal{T}$. By Lemma 2.1, the absolute value of the error $\delta(x,y)$ between x and y can be bounded by $\sum_v 2\varepsilon d(v,\alpha(v))$ where the sum is over all nodes v on the paths connecting x and y to z in $\mathcal{T}$. The next lemma brings the sum of the distances $d(v,\alpha(v))$ on a path on $\mathcal{T}$ into a more suitable form for our analysis. The lemma states the sum in terms of values β_v for every node v, which are defined as follows: $\beta_v := (v|r)_{\alpha(v)}/(\alpha(v)|r)_v$. Hence β_v is the ratio of the distances of $\alpha(v)$ and v to the Steiner node t_v respectively.

LEMMA 2.2. *Let $x, y \in V$ be two nodes such that y is an ancestor of x in $\mathcal{T}$. Let $x = x_1, \dots, x_k = y$ be the path connecting x and y in $\mathcal{T}$. Let $t_0 = x$ and $t_i = t_{x_i}$ for $i \ge 1$. We have*

$$\begin{aligned} \sum_{i=1}^{k-1} d(x_i, x_{i+1}) &= \sum_{i=1}^{k-1} d(x_i, \alpha(x_i)) \\ &\le \sum_{i=1}^{k-1} T(t_{i-1}, t_i) \cdot \left(1 + 2 \cdot \sum_{j=i}^{k-1} \prod_{\ell=i}^{j} \beta_{x_\ell}\right). \end{aligned} \tag{4}$$

PROOF. We prove that

$$T(x_i, t_i) \ = \ T(t_{i-1}, t_i) + \sum_{j=1}^{i-1} T(t_{j-1}, t_j) \cdot \prod_{\ell=j}^{i-1} \beta_{x_\ell} \tag{5}$$

by induction on i. Equation (5) holds for $i = 1$ because $T(x_1, t_1) = T(t_0, t_1)$. For $i > 1$, we have

$$\begin{aligned} &T(x_i, t_i) \\ &\quad= T(t_{i-1}, t_i) + T(x_i, t_{i-1}) \\ &\quad= T(t_{i-1}, t_i) + \beta_{x_{i-1}} \cdot T(x_{i-1}, t_{i-1}) \\ &\quad= T(t_{i-1}, t_i) + \beta_{x_{i-1}} \left(T(t_{i-2}, t_{i-1}) + \sum_{j=1}^{i-2} T(t_{j-1}, t_j) \prod_{\ell=j}^{i-2} \beta_{x_\ell} \right) \\ &\quad= T(t_{i-1}, t_i) + \sum_{j=1}^{i-1} T(t_{j-1}, t_j) \cdot \prod_{\ell=j}^{i-1} \beta_{x_\ell}. \end{aligned}$$

The second equation follows by definition of $\beta_{x_{i-1}}$, the third equation uses the induction hypothesis. Given Equation (5) and using that $d(x_i, x_{i+1}) = (1 + \beta_{x_i}) T(x_i, t_i)$, we can now prove the lemma as follows:

$$\begin{aligned} &\sum_{i=1}^{k-1} d(x_i, x_{i+1}) \\ &\quad= \sum_{i=1}^{k-1} (1 + \beta_{x_i}) T(x_i, t_i) \\ &\quad= \sum_{i=1}^{k-1} (1 + \beta_{x_i}) \cdot \left(T(t_{i-1}, t_i) + \sum_{j=1}^{i-1} T(t_{j-1}, t_j) \cdot \prod_{\ell=j}^{i-1} \beta_{x_\ell} \right) \\ &\quad= \sum_{i=1}^{k-1} T(t_{i-1}, t_i) \cdot \sum_{j=i}^{k-1} (1 + \beta_{x_j}) \cdot \prod_{\ell=j+1}^{k-1} \beta_{x_\ell} \\ &\quad\le \sum_{i=1}^{k-1} T(t_{i-1}, t_i) \cdot \left(1 + 2 \cdot \sum_{j=i}^{k-1} \prod_{\ell=i}^{j} \beta_{x_\ell} \right). \end{aligned}$$

□

From Lemmas 2.1 and 2.2, we see that the error $\delta(x,y)$ between two nodes x and y depends on the values of β_v for all the nodes on the path connecting x and y in $\mathcal{T}$ as well as on the number of nodes on this path. We have some flexibility which allows us to play with the values of β_v as well as the length of the paths on $\mathcal{T}$. In Line 7 of Algorithm 1, we remove q from the node set and recursively construct a tree on the remaining nodes. The Gromov product $(p|q)_r$ is symmetric in p and q. We could therefore change the roles of p and q in our algorithm and recursively construct the tree for $V \setminus \{p\}$ instead of $V \setminus \{q\}$. Like this, it is possible to guarantee that all $\beta_v \le 1$ by always removing the node from V which is farther away from r. It can be shown that by doing this,

Algorithm 2 Low Distortion Tree Construction

select arbitrary root node $r \in V$;
for all $v \in V$ **do** $n_v := 1$ **od**;
return tree $T := \mathbf{constructTree}(V \setminus \{r\}, r)$;

function constructTree(V,r):
  if $|V| > 3$ **then**
    choose $p, q \in V$ s.t. $(p|q)_r = \max_{u,v \in V}\{(u|v)_r\}$ and $\left(\frac{(q|r)_p}{(p|r)_q} \le \frac{1}{\lambda}\right) \vee \left(\frac{(q|r)_p}{(p|r)_q} < \lambda \wedge n_p \ge n_q\right)$;
    $n_p := n_p + n_q$;
    // $\alpha(q) := p$; $\beta_q := (q|r)_p/(p|r)_q$;
    $T := \mathbf{constructTree}(V \setminus \{q\}, r)$;
    add Steiner node t_q at distance $(q|r)_p$ from p;
    add node q at distance $(p|r)_q$ from t_q
  else
    T is line segment from $v \in V$ to r;
    // $\alpha(v) := \alpha(r) := r$; $\beta_v := \beta_r := 0$
  fi;
  return T
end constructTree

the resulting tree has distortion $(1+\varepsilon)^{\Theta(n)}$. If we always remove the node which belongs to the smaller subtree in the final tree, it is also possible to reduce path lengths on $\mathcal{T}$ to $\mathrm{O}(\log n)$. If we could guarantee that paths on $\mathcal{T}$ have length $\mathrm{O}(\log n)$ and that all $\beta_v \le 1$, the resulting tree would have distortion $(1+\varepsilon)^{\mathrm{O}(\log n)}$. However, it is not possible to have $\beta_v \le 1$ and short paths on $\mathcal{T}$ at the same time.

Our solution is to introduce a parameter $\lambda > 1$. If p and q are such that we either get $\beta_p \le 1/\lambda$ or $\beta_q \le 1/\lambda$, we remove the node farther away from r. The sum of Lemma 2.2 then behaves like a geometric series and we do not care about the number of terms (i.e., the path lengths in $\mathcal{T}$). If we cannot get $\beta_p \le 1/\lambda$ or $\beta_q \le 1/\lambda$, we remove the node belonging to the smaller subtree and try to keep $\mathcal{T}$ as balanced as possible. The details of this are given by Algorithm 2. The following lemma formally states the role of the parameter λ.

LEMMA 2.3. *Let z be the least common ancestor of two nodes x and y in $\mathcal{T}$ and let $x = x_1, \dots, x_{k_x} = z$ and $y = y_1, \dots, y_{k_y} = z$ be the paths in $\mathcal{T}$ connecting x and y with z, respectively. At most $\log n$ of the values β_{x_i} and at most $\log n$ of the values β_{y_j} are larger than $1/\lambda$. All β-values are less than λ.*

PROOF. Let us look at an execution of **constructTree**(V, r). In order to have $\beta_q > 1/\lambda$, we need that $n_p \ge n_q$. Before the recursive call of **constructTree**, n_p is set to $n_p + n_q$. Thus, in the end, $n_p = n_{\alpha(q)} \ge 2n_q$. Because for all p, $n_p \le n$, there are at most $\log n$ nodes u with $\beta_u > 1/\lambda$ on the path from q to r in the tree $\mathcal{T}$. $\square$

By combining Lemmas 2.1–2.3, we can now bound $\delta(x,y)$ from below as shown by Lemma 2.4.

LEMMA 2.4. *The error $\delta(x,y)$ between the tree distance $T(x,y)$ and $d(x,y)$ is bounded from below by*

$$\begin{aligned}\delta(x,y) &= T(x,y) - d(x,y)\\ &\ge -4 \cdot \varepsilon \cdot \lambda^{\log n} \cdot \left(\log n + \frac{\lambda}{\lambda - 1}\right) \cdot T(x,y).\end{aligned}$$

PROOF. Let z be the least common ancestor of x and y in $\mathcal{T}$ and let $x = x_1, \dots, x_{k_x} = z$ and $y = y_1, \dots, y_{k_y} = z$ be the paths in $\mathcal{T}$ connecting x and y with z, respectively. Lemma 2.1 implies that

$$\delta(x,y) \ge -2\varepsilon \cdot \left(\sum_{i=1}^{k_x-1} d(x_i, x_{i+1}) + \sum_{j=1}^{k_y-1} d(y_j, y_{j+1})\right).$$

We can bound the two sums in the above inequality by using Lemma 2.2. By Lemma 2.3, the number of β_{x_i} that are larger than $1/\lambda$ is at most $\log n$. Applying this and the fact that $\beta_u \le \lambda$ for all nodes u, we obtain the following estimate for the first sum of the above inequality where $t_0 = x_1$ and $t_i = t_{x_i}$ for $i \ge 1$:

$$\begin{aligned}\sum_{i=1}^{k_x-1} d(x_i, x_{i+1}) &\le \sum_{i=1}^{k-1} T(t_{i-1}, t_i) \cdot \left(1 + 2 \cdot \sum_{j=i}^{k-1} \prod_{\ell=i}^{j} \beta_{x_\ell}\right)\\ &\le \sum_{i=1}^{k-1} T(t_{i-1}, t_i) \cdot 2\lambda^{\log n} \left(\log n + \frac{1}{1 - \frac{1}{\lambda}}\right)\end{aligned}$$

The second of the distances $d(y_j, y_{j+1})$ can be bounded analogously and the lemma now follows because we can write the tree distance $T(x,y)$ as follows where t_i is defined as before and $s_0 = y$ and $s_j = t_{y_j}$ for $j \ge 1$:

$$\begin{aligned}T(x,y) &= T(x, t_{x_{k_x-1}}) + T(t_{x_{k_x-1}}, t_{y_{k_y-1}}) + T(t_{y_{k_y-1}}, y)\\ &= \sum_{i=1}^{k_x-1} T(t_{i-1}, t_i) + T(t_{x_{k_x-1}}, t_{y_{k_y-1}})\\ &+ \sum_{j=1}^{k_y-1} T(s_{j-1}, s_j).\end{aligned}$$

$\square$

For $\varepsilon \ll 1/\log n$, we can obtain a useful upper bound on $\delta(x,y)$ in the same way as the lower bound given by Lemma 2.4. For larger ε, bounding $\delta(x,y)$ from above is less straight-forward, but we can also do it as shown by the next lemma.

LEMMA 2.5. *For $\lambda \ge (1+2\varepsilon)/(1-2\varepsilon)$, the error $\delta(x,y)$ between the tree distance $T(x,y)$ and $d(x,y)$ is bounded from above by*

$$\begin{aligned}\delta(x,y) &= T(x,y) - d(x,y)\\ &\le 4\varepsilon \cdot \frac{A_\lambda \cdot (\lambda + 2(1+\lambda)\varepsilon)^{2\log n - 1} \cdot \log n}{(1 - \varepsilon A_\lambda)^{2\log n}} \cdot d(x,y)\end{aligned}$$

for $A_\lambda = \frac{2\lambda^2}{\lambda - 1}$.

PROOF. As before, let z be the least common ancestor of x and y in $\mathcal{T}$ and let $x = x_1, \dots, x_{k_x} = z$ and $y = y_1, \dots, y_{k_y} = z$ be the paths in $\mathcal{T}$ connecting x and y with z respectively. Let K be the number of values in $\{\beta_{x_1}, \dots, \beta_{x_{k_x-1}}\} \cup \{\beta_{y_1}, \dots, \beta_{y_{k_y-1}}\}$ that are larger than $1/\lambda$. We prove the following inequality for $K \ge 1$ by induction on K:

$$\delta(x,y) =\le 2\varepsilon \cdot \frac{A_\lambda \cdot (\lambda + 2(1+\lambda)\varepsilon)^{K-1} \cdot K}{(1 - 2\varepsilon A_\lambda)^K} \cdot d(x,y) \tag{6}$$

The lemma then follows by Lemma 2.3. Let us first prove Inequality (6) directly for $K = 1$. For a set of β-values $\{\beta_1, \dots, \beta_k\}$, we have

$$\left(1 + 2 \cdot \sum_{j=i}^{k-1} \prod_{\ell=i}^{j} \beta_\ell\right) \le \frac{2\lambda}{1 - 1/\lambda} = A_\lambda \tag{7}$$

if at most one $\beta_i \in [1/\lambda, \lambda]$ and all other β-values $\beta_i \le 1/\lambda$. We can then bound $\delta(x,y)$ by using the same techniques as in the proof of Lemma 2.4, and obtain

$$\begin{aligned} T(x,y) - d(x,y) &\leq 2\varepsilon A_\lambda \cdot T(x,y) \\ \implies \delta(x,y) &\leq 2\varepsilon \cdot \frac{A_\lambda}{1-2\varepsilon A_\lambda} \cdot d(x,y) \end{aligned}$$

for $K = 1$. Note that the above Inequality also holds if $K = 0$ and therefore the lemma also holds in this case.

For the induction hypothesis, let u be the first node such that $u \in \{x_1, \ldots, x_{k_x-1}\} \cup \{y_1, \ldots, y_{k_y-1}\}$ with $\beta_u > 1/\lambda$ that is removed from V when constructing the tree T. W.l.o.g., we can assume that u is on the path from x to z. Let us therefore assume that $u = x_\mu$. Let y_ν be the node closest to y of the nodes y_i remaining in V when x_μ is removed. By using Lemma 2.1, we can now bound $\delta(x,y)$ as follows:

$$\delta(x,y) \leq \delta(x_{\mu+1}, y_\nu) + 2\varepsilon \cdot \left(\sum_{i=1}^{\mu} d(x_i, x_{i+1}) + \sum_{j=1}^{\nu-1} d(y_j, y_{j+1}) \right).$$

We can use the induction hypothesis to bound $\delta(x_\mu, y_\nu)$ and Lemma 2.2 to bound the sums over $d(x_i, x_{i+1})$ and $d(y_j, y_{j+1})$. Setting $t_0 = x$, $t_i = t_{x_i}$ for $i \geq 1$, $s_0 = y$ and $s_i = t_{y_i}$ for $i \geq 1$ gives

$$\begin{aligned} &\delta(x,y) \\ &\leq 2\varepsilon \cdot \frac{A_\lambda \cdot (\lambda + 2(1+\lambda)\varepsilon)^{K-2} \cdot (K-1)}{(1-2\varepsilon A_\lambda)^{K-1}} \cdot d(x_{\mu+1}, y_\nu) \\ &\quad + 2\varepsilon \cdot \sum_{i=1}^{\mu} T(t_{i-1}, t_i) \cdot \left(1 + 2 \cdot \sum_{j=i}^{\mu} \prod_{\ell=i}^{j} \beta_{x_\ell} \right) \\ &\quad + 2\varepsilon \cdot \sum_{i=1}^{\nu-1} T(s_{i-1}, s_i) \cdot \left(1 + 2 \cdot \sum_{j=i}^{\nu-1} \prod_{\ell=i}^{j} \beta_{y_\ell} \right) \\ &\leq 2\varepsilon \cdot \frac{A_\lambda \cdot (\lambda + 2(1+\lambda)\varepsilon)^{K-2} \cdot (K-1)}{(1-2\varepsilon A_\lambda)^{K-1}} \cdot d(x_{\mu+1}, y_\nu) \\ &\quad + 2\varepsilon A_\lambda \cdot T(x,y). \end{aligned} \tag{8}$$

In order to obtain something useful from Inequality 8, we need to bound $d(x_{\mu+1}, y_\nu)$. Consider an execution of **constructTree**(V, r) where q is removed from V, let $p = \alpha(q)$, and let w be a node in V when q is removed. By using the ε-4PC, we get the following bound on $d(p,w)$:

$$\begin{aligned} d(p,w) &\leq d(q,w) + d(p,r) - d(q,r) + 2\varepsilon \cdot d(p,q) \\ &= d(q,w) + (\beta_q - 1) \cdot (p|r)_q + 2\varepsilon(1+\beta_q) \cdot (p|r)_q. \end{aligned}$$

By the maximality condition in Line 7 of the algorithm, we have

$$d(p,r) + d(q,r) - d(p,q) \geq d(w,r) + d(q,r) - d(w,q)$$

and therefore

$$\begin{aligned} &d(q,w) \\ &\geq (w|r)_q \\ &= \frac{1}{2} \cdot \big(d(w,q) + d(q,r) - d(w,r)\big) \\ &\geq \frac{1}{2} \cdot \big(d(p,q) + d(q,r) - d(p,r)\big) \\ &= (p|r)_q. \end{aligned}$$

We can therefore bound $d(p,w)$ as

$$d(p,w) \leq \big(\beta_q + 2\varepsilon(1+\beta_q)\big) \cdot d(q,w).$$

For $\beta_q \leq 1/\lambda$, we get

$$\begin{aligned} d(p,w) &\leq \left[\frac{1}{\lambda} + 2\varepsilon \cdot \left(1 + \frac{1}{\lambda}\right) \right] \cdot d(q,w) \\ &\leq \left[\frac{1-2\varepsilon}{1+2\varepsilon} + 2\varepsilon \cdot \frac{2}{1+2\varepsilon} \right] \cdot d(q,w) \ \leq \ d(q,w) \end{aligned}$$

since we assume that $\lambda \geq (1+2\varepsilon)/(1-2\varepsilon)$. For $\beta \leq \lambda$, we have $d(p,w) \leq (\lambda + 2\varepsilon(1+\lambda)) \cdot d(q,w)$. Because all β_{x_i} and β_{y_j} for $i < \mu$ and $j < \nu$ are less than $1/\lambda$, we have $d(x_\mu, y_\nu) \leq d(x,y)$ and therefore $d(x_{\mu+1}, y_\nu) \leq (\lambda + 2(1+\lambda)\varepsilon) d(x,y)$. Plugging this into Inequality (8) yields

$$\begin{aligned} &(1 - 2\varepsilon A_\lambda) \cdot T(x,y) \ \leq \\ &\quad \left(1 + 2\varepsilon \cdot \frac{A_\lambda \cdot (\lambda + 2(1+\lambda)\varepsilon)^{K-1} \cdot (K-1)}{(1-2\varepsilon A_\lambda)^{K-1}} \right) \cdot d(x,y) \end{aligned}$$

and therefore

$$\begin{aligned} &T(x,y) \ \leq \\ &\quad \left(1 + 2\varepsilon \cdot \frac{A_\lambda + A_\lambda(\lambda + 2(1+\lambda)\varepsilon)^{K-1}(K-1)}{(1-2\varepsilon A_\lambda)^{K}} \right) \cdot d(x,y). \end{aligned}$$

This concludes the proof. □

Based on Lemmas 2.4 and 2.5, it is now possible to prove the main theorem of this section.

THEOREM 2.6. *Let (V,d) be a metric space that satisfies the ε-4PC for a given $\varepsilon \in [0,1]$. There is a constant κ, such that it is possible to embed (V,d) into a tree with distortion at most $(1+\varepsilon)^{\kappa \cdot \log n}$ where $n = |V|$.*

PROOF. We set

$$\lambda = \max \left\{ 1 + \frac{1}{\log n}, \frac{1+2\varepsilon}{1-2\varepsilon} \right\}.$$

If $\varepsilon < 1/(2A_\lambda) - \xi$ for a constant ξ, the theorem then follows from Lemmas 2.4 and 2.5. For larger ε, the lemma follows because any metric space can be embedded into a tree with distortion n. □

3. LOWER BOUNDS

In this section, we show that the upper bound given by Theorem 2.6 is essentially tight. We also extend a lower bound from [12] for distance labelings of hyperbolic metric spaces to metric spaces satisfying the ε-4PC. Both lower bounds are based on a construction related to the pyramid construction for graphs introduced in [12]. In analogy, we call our construction the pyramid of a metric space:

DEFINITION 3.1. *Let $\mathcal{X} = (X,d)$ be a metric space. For a parameter $\delta \in [0,1]$, the pyramid $\mathcal{P}_\delta(\mathcal{X})$ of $\mathcal{X}$ is a space (X, d') with a new distance function $d'(x,y) := (1+\delta)^{\log d(x,y)}$ for $x, y \in X$ and $d'(x,x) := 0$ for $x \in X$.*

We want to show that for every metric $\mathcal{X}$ and every $\delta \in [0,1]$, the pyramid $\mathcal{P}_\delta(\mathcal{X})$ is a metric space that satisfies the ε-4PC for $\varepsilon \in \mathrm{O}(\delta)$. In order to prove this, we need the following lemma.

LEMMA 3.1. *For $A \geq B \geq 0$ and $0 \leq \delta \leq 1$, we have*

$$(1+\delta)^{\log(A+B)} - (1+\delta)^{\log A} \leq \frac{e-1}{\ln 2} \cdot \delta \cdot (1+\delta)^{\log B}.$$

PROOF. Let $\lambda := B/A \leq 1$. We then get

$$\begin{aligned}
&(1+\delta)^{\log(A+B)} - (1+\delta)^{\log A} \\
&= (1+\delta)^{\log A} \cdot \left[(1+\delta)^{\log(1+\lambda)} - 1\right] \\
&\leq (1+\delta)^{\log A} \cdot \left[e^{\delta \log(1+\lambda)} - 1\right] \\
&\leq (1+\delta)^{\log A} \cdot (e-1)\delta \log(1+\lambda) \\
&\leq \frac{e-1}{\ln 2} \cdot \delta \cdot (1+\delta)^{\log \lambda} \cdot (1+\delta)^{\log A} \\
&= \frac{e-1}{\ln 2} \cdot \delta \cdot (1+\delta)^{\log B}.
\end{aligned}$$

The second inequality uses the fact that $e^x - 1 \leq (e-1)x$ for $0 \leq x \leq 1$, the third inequality follows because

$$\log(1+\lambda) \leq \frac{\lambda}{\ln 2} \leq \frac{\lambda^{\log(1+\delta)}}{\ln 2} = \frac{(1+\delta)^{\log \lambda}}{\ln 2}.$$

Note that $0 \leq \lambda \leq 1$ and $0 \leq \log(1+\delta) \leq 1$. □

LEMMA 3.2. *For every $\delta \in [0,1]$ and every metric space $\mathcal{X} = (X,d)$, the pyramid $\mathcal{P}_\delta(\mathcal{X}) = (X,d')$ is a metric space that satisfies the ε-4PC for $\varepsilon = (e-1)/\ln 2 \cdot \delta$*

PROOF. If $F : \mathbb{R}^+ \to \mathbb{R}^+$ is a non-decreasing, concave and $F(0) = 0$ then for any metric space (X,d), the metric transform $(X, F \circ d)$ is also a metric space (Where $(F \circ d)(u,v) = F(d(u,v))$). Therefore $\mathcal{P}_\delta(\mathcal{X})$ is a metric space since it is the metric transform induced by $F(r) = 0$ if $r = 0$ and $F(r) = r^{\log(1+\delta)}$ otherwise. If is easy to check that F is indeed non-decreasing and concave.

To prove that $\mathcal{P}_\delta(\mathcal{X})$ satisfies the ε-4PC consider four points $w,x,y,z \in X$. W.l.o.g., assume that

$$d'(w,x)+d'(y,z) \geq \max\{d'(w,y)+d'(x,z), d'(w,z)+d'(x,y)\}$$

and that

$$d'(x,z) = \min\{d'(w,y), d'(x,z), d'(w,z), d'(x,y)\}. \quad (9)$$

We have

$$\begin{aligned}
&d'(w,x)+d'(y,z) - \max\{d'(w,y)+d'(x,z), d'(w,z)+d'(x,y)\} \\
&\leq d'(w,x) + d'(y,z) - \left(d'(w,z) + d'(x,y)\right) \\
&= (1+\delta)^{\log d(w,x)} + (1+\delta)^{\log d(y,z)} \\
&\quad - \left((1+\delta)^{\log d(w,z)} + (1+\delta)^{\log d(x,y)}\right) \\
&\leq \frac{e-1}{\ln 2} \cdot \delta \cdot \left((1+\delta)^{\log(d(w,x)-d(w,z))}\right. \\
&\quad \left.+ (1+\delta)^{\log(d(y,z)-d(x,y))}\right) \\
&\leq 2 \cdot \frac{e-1}{\ln 2} \cdot \delta \cdot (1+\delta)^{\log d(x,z)} \\
&= 2 \cdot \frac{e-1}{\ln 2} \cdot \delta \cdot d'(x,z).
\end{aligned}$$

The second inequality follows by the triangle inequality in the metric space $\mathcal{X}$ and by Lemma 3.1. Note that we need Assumption (9) to be able to apply Lemma 3.1. □

THEOREM 3.3. *Consider the metric $\mathcal{X} = (X,d)$ with $X = \{x_1, \ldots, x_n\}$ and $d(x_i,x_j) = |j-i|$. There is a constant κ such that every tree embedding of $\mathcal{P}_\delta(\mathcal{X})$ has distortion larger than $(1+\varepsilon)^{\kappa \cdot \log n}$ for $\varepsilon = (e-1)/\ln 2 \cdot \delta$.*

PROOF. For the sake of contradiction assume that there is a tree T into which $\mathcal{P}_\delta(\mathcal{X})$ embeds with distortion at most $(1+\varepsilon)^{\kappa \cdot \log n}$. Let $T(x_i,x_j)$ be the distance between x_i and x_j on T. W.l.o.g., we can assume that the distances on T dominate the distances in $\mathcal{P}_\delta(\mathcal{X})$, that is for all $1 \leq i < j \leq n$,

$$\begin{aligned}
&d(x_i,x_j) \leq T(x_i,x_j) \leq (1+\varepsilon)^{\kappa \cdot \log n} \cdot d(x_i,x_j) \\
&\leq (1+\delta)^{\kappa' \cdot \log n} \cdot d(x_i,x_j) \quad \text{for } \kappa' = \frac{e-1}{\ln 2} \cdot \kappa.
\end{aligned} \quad (10)$$

Let r be a node (r can be a Steiner node) that splits T into two subtrees T_1 and T_2. We call i a switching index with respect to T_1 and T_2 if $x_i \in T_\theta$ and $x_{i+1} \in T_{1-\theta}$ for $\theta \in \{0,1\}$. Assume that there are two switching indices i and j with $j - i \geq n^{\kappa'} + 1$. The tree distance between two nodes $u \in T_1$ and $v \in T_2$ is $T(u,v) = T(u,r) + T(v,r)$. For two nodes u and v in the same subtree, we have $T(u,v) \leq T(u,r) + T(v,r)$. If x_{i+1} and x_j are in the same subtree, we obtain a contradiction to Inequality (10) because

$$\begin{aligned}
&T(x_{i+1},r) + T(x_j,r) + T(x_i,r) + T(x_{j+1},r) \\
&\geq T(x_{i+1},x_j) + T(x_i,x_{j+1}) \\
&\geq d(x_{i+1},x_j) + d(x_i,x_{j+1}) > 2(1+\delta)^{\kappa' \log n}
\end{aligned}$$

and

$$\begin{aligned}
&T(x_i,r) + T(x_{i+1},r) + T(x_j,r) + T(x_{j+1},r) \\
&= T(x_i,x_{i+1}) + T(x_j,x_{j+1}) \\
&\leq (1+\delta)^{\kappa' \log n} \cdot \left(d(x_i,x_{i+1}) + d(x_j,x_{j+1})\right) \\
&= 2(1+\delta)^{\kappa' \log n}.
\end{aligned}$$

In the other case where x_i and x_j are in the same subtree, a contradiction can be obtained analogously. We can therefore assume that there is no split into two subtrees with switching indices that are $n^{\kappa'} + 1$ apart.

In the following, we assume for simplicity that all points $x_i \in X$ correspond to leaves of T and that all inner nodes of T are Steiner nodes of degree 3. By allowing edges of length 0, we can bring any tree to a tree of this form without changing the distances between any two nodes $x_i, x_j \in X$. We consider the size of a subtree of T to be the number of leaves in the subtree. We first prove that no Steiner node partitions the T into three large subtrees. In particular, if T_1, T_2, and T_3 are the three subtrees of a given Steiner node t, the size of the smallest of the three subtrees is less than $n^{\kappa'} + 1$. Let us call subtrees with at least $n^{\kappa'} + 1$ nodes heavy and subtrees with less than $n^{\kappa'} + 1$ nodes light. For the sake of contradiction assume that all three subtrees of t are heavy. Let i be the smallest index such that x_i and x_{i+1} belong to two different subtrees and let j be the largest index such that x_j and x_{j+1} belong to two different subtrees. If all three subtrees are heavy, we have $j - i \geq n^{\kappa'} + 1$. Clearly, two of the four nodes x_i, x_{i+1}, x_j, and x_{j+1} belong to the same subtree. W.l.o.g., let us assume it is T_1. Indices i and j then are switching indices with respect to the subtrees T_1 and $T_2 \cup T_3$ and we have a contradiction.

Let us now have a look at all Steiner nodes which have two heavy subtrees and show that all these nodes lie on a single path P. Consider three such nodes u, v, and w and assume that they do not lie on a path. Then there is a node t that splits T into three subtrees T_u, T_v, and T_w containing nodes u, v, and w, respectively. However, since u, v, and w all have two heavy subtrees, T_u, T_v, and T_w are all heavy. But, this is not possible because t can only have two heavy subtrees.

All nodes that are not on the path P are either in the light subtrees of the nodes of P or in one of the two heavy subtrees of size less than $2(n^{\kappa'}+1)$ hanging from the end nodes of P. We now partition P into k segments $P_1, \ldots, P_k$ such that the combined size of the subtrees of each segment is between $n^{2\kappa'}+1$ and $2(n^{2\kappa'}+1)$. Let u_s for $s \in [k-1]$ be the last node of P belonging to segment P_s, that is, node u_s splits T into segments $\{P_1, \ldots, P_s\}$ and $\{P_{s+1}, \ldots, P_k\}$.

The next step is to get a lower bound on the tree distance between nodes u_s and u_{s+1} separating consecutive segments. We partition the nodes x_i into three sets A, B, and C such that A contains all nodes in segments $\{P_1, \ldots, P_{s-1}\}$, B contains the nodes of segment P_s, and C contains the nodes in segments $\{P_{s+1}, \ldots, P_k\}$. Let i be the smallest index for which x_i and x_{i+1} belong to different sets. and let j be the largest index for which x_j and x_{j+1} belong to different sets. Because all sets have size at least $n^{2\kappa'}+1$, we have $j-i \geq n^{2\kappa'}+1$. Two of the nodes belong to the same subset. It is not possible that two nodes belong to A because in that case i and j are switching indices with respect to the partition $(A, B\cup C)$ of T at node u_s. Similarly, it is not possible that two nodes belong to C and we therefore have two nodes in B, one node in A, and one node in C. Assume that $x_i \in A$, $x_{i+1}, x_j \in B$, and $x_{j+1} \in C$. All other cases are analogous. We know that $T(x_i, x_{i+1}) \leq (1+\delta)^{\kappa' \log n}$. Because u_s lies on the path connecting x_i and x_{i+1}, one of the distances $T(u_s, x_i)$ and $T(u_s, x_{i+1})$ is at most half of $T(x_i, x_{i+1})$. Assume that $T(u_s, x_{i+1}) \leq (1+\delta)^{\kappa' \log n}/2$ and similarly that $T(u_{s+1}, x_j) \leq (1+\delta)^{\kappa' \log n}/2$. We then have

$$\begin{aligned}(1+\delta)^{2\kappa' \log n} &\leq d(x_{i+1}, x_j) \leq T(x_{i+1}, x_j) \\ &= T(x_{i+1}, u_s) + T(u_s, u_{s+1}) + T(u_{s+1}, x_j) \\ &\leq T(u_s, u_{s+1}) + (1+\delta)^{\kappa' \log n}.\end{aligned}$$

We therefore obtain $T(u_s, u_{s+1}) \geq (1+\delta)^{2\kappa' \log n} - (1+\delta)^{\kappa' \log n}$. There is a node x of the first segment P_1 for which $T(x, u_1) \geq 1/2$ and there is a node y of the last segment P_k for which $T(y, u_{k-1}) \geq 1/2$. We have

$$\begin{aligned}T(x,y) &= T(x, u_1) + \sum_{s=1}^{k-2} T(u_s, u_{s+1}) + T(u_{k-1}, y) \\ &\geq 1 + (k-2) \cdot \left[(1+\delta)^{2\kappa' \log n} - (1+\delta)^{\kappa' \log n}\right].\end{aligned}$$

The distortion of T therefore is at least

$$\max_{x,y \in X} \frac{T(x,y)}{d(x,y)} \geq \frac{1 + (k-2)\left[(1+\delta)^{2\kappa' \log n} - (1+\delta)^{\kappa' \log n}\right]}{(1+\delta)^{\log n}}. \tag{11}$$

In order to get a contradiction, we want the right-hand side of Inequality (11) to be larger than $(1+\delta)^{\kappa' \log n}$. To make the right-hand side large enough, we need

$$k - 2 > \frac{1}{(1+\delta)^{\kappa'}} \cdot \frac{(1+\delta)^{(1+\kappa')\log n} - 1}{(1+\delta)^{\kappa' \log n} - 1}.$$

By the way, we partition P into segments, we have $k-2 > (1/2 - \mathrm{o}(1))n^{1-2\kappa'}$. We further get

$$\begin{aligned}&\frac{(1+\delta)^{(1+\kappa')\log n} - 1}{(1+\delta)^{\kappa' \log n} - 1} \\ &= \sum_{i=0}^{\lfloor 1/\kappa' \rfloor} (1+\delta)^{(1-i\kappa')\log n} + \frac{(1+\delta)^{(1-\kappa'\lfloor 1/\kappa' \rfloor)\log n} - 1}{(1+\delta)^{\kappa' \log n} - 1} \\ &< \left(2 + \frac{1}{\kappa'}\right) \cdot (1+\delta)^{\log n}.\end{aligned}$$

We therefore obtain a contradiction to Inequality (10) if

$$\left(\frac{1}{2} - \mathrm{o}(1)\right) n^{1-2\kappa'} \geq \left(2 + \frac{1}{\kappa'}\right) \cdot (1+\delta)^{(1-\kappa')\log n},$$

which is true for $\kappa' \leq \frac{1-\log(1+\delta)}{2-\log(1+\delta)} - \mathrm{o}(1)$. This is a constant for all values of δ for which $\varepsilon < 1$, that is, for $\delta < \ln 2/(e-1)$. $\square$

Since ε-4PC is stronger than assuming that every 4 points of a metric embed into a tree with distortion $1 + \mathrm{O}(\varepsilon)$, Theorem 3.3 implies that there is a metric space that cannot be embedded into a tree with distortion $(1+\varepsilon)^{\kappa \log n}$ for some constant κ although every 4-point submetric has a $(1+\varepsilon)$-distortion tree embedding. This can be generalized to a condition on k points.

THEOREM 3.4. *There is a constant $\kappa > 0$ such that for every $\varepsilon \in [0,1]$ there is a metric space $\mathcal{X}_\varepsilon = (X, d)$ such that every k points of $\mathcal{X}_\varepsilon$ embeds into a tree metric with distortion $1+\varepsilon$ but such that embedding $\mathcal{X}_\varepsilon$ into a tree requires distortion $(1+\varepsilon)^{\kappa \log_k(|X|)}$.*

PROOF. Let $\mathcal{X} = (X, d')$ be the shortest path metric of a path of length n. We consider the pyramid $\mathcal{P}_{c\delta/\log_k}(\mathcal{X})$ for some constant c. The metric $\mathcal{P}_{c\delta/\log_k}(\mathcal{X})$ satisfies ε-4PC for $\varepsilon = c\cdot(e-1)/\ln 2 \cdot \delta/\log k$. For suitable choice of c, every k points of $\mathcal{P}_{c\delta/\log_k}(\mathcal{X})$ can therefore be embedded into a tree with distortion $1+\varepsilon$ by Theorem 2.6. The proof now follows from Theorem 3.3. $\square$

A similar result to Theorem 3.4 has been proven in [3]. There it has been shown that there is a metric space for which every tree embedding has distortion $\Omega((1+\varepsilon)^{\lceil \log_k n \rceil})$ but where every k-point submetric embeds into a tree metric with distortion $1+\varepsilon$. The statement of Theorem 3.4 is strictly stronger than the result of [3] if $\varepsilon \leq d/\log n$ for some constant d.

To conclude our section on lower bounds, we show how the techniques developed in this section can be used to extend a distance labeling lower bound for hyperblic metrics described in [12]. In [12], it is shown by Gavoille and Ly that there is a family of δ-hyperbolic metrics where every $\log(2k)$-multiplicative distance labeling scheme requires labels of $\Omega(n^{1/k})$ bits. They define the pyramid graph $\overline{G}$ of an unweighted graph G and show that $\overline{G}$ is hyperbolic. The family for the distance labeling lower bound is then obtained by considering the pyramids of all connected subgraphs of $G_{n,k}$ where $G_{n,k}$ is a graph with n nodes, girth k, and maximal number of edges. By applying our pyramid construction from Definition 3.1 to the shortest path metrics of all connected subgraphs of $G_{n,k}$, we obtain the following lower bound by using the same argumentation as in [12].

THEOREM 3.5. *For all $\varepsilon \in [0,1]$, $n \geq 1$, and $k \geq 1$, there is a family of metrics satisfying ε-4PC for which every $(1+\varepsilon)^{\log k}$-approximate distance labeling scheme needs labels of $n^{\Omega(1/k)}$ bits. In particular, every distance labeling with polylogarithmic labels has stretch $(1+\varepsilon)^{\Omega(\log\log n)}$.*

Since trees have exact distance labelings with labels of size $\mathrm{O}(\log^2 n)$ [13, 24], Theorem 3.5 in particular implies that even by using a polylogarithmic number of trees, ε-4PC metrics cannot be approximated better than by a factor of $(1+\varepsilon)^{\Omega(\log\log n)}$.

4. TREE EMBEDDINGS WITH ADDITIVE DISTORTION

In his seminal work on hyperbolic metric spaces, Gromov shows the following result:

LEMMA 4.1. **[14]** *Let (X, d) be a metric and let $r \in X$ be an arbitrary point of this metric. If for every sequence $x = x_1, \ldots, x_t = y$ of points $x_i \in X \setminus \{r\}$, $(x|y)_r \geq \min_{1 \leq i < t}(x_i|x_{i+1})_r - \Delta$, (X, d) embeds into a tree metric with additive distortion at most 2Δ.*

By the definition of the Gromov product, we also directly get that $(x|y)_r \geq \min_{1 \leq i < t}(x_i|x_{i+1})_r - 6\Delta$ if a given metric embeds into a tree with additive distortion 2Δ. Gromov's tree embedding thus gives a 6-approximation for the problem of finding a tree embedding with minimum additive distortion. With a more careful analysis, it is even possible to show that $(x|y)_r \geq \min_{1 \leq i < t}(x_i|x_{i+1})_r - 3\Delta$ if the given metric embeds into a tree with additive distortion 2Δ. We therefore even obtain a 3-approximation for the minimum additive distortion problem. Independently, a 3-approximation for the problem of minimizing the additive distortion of a tree embedding has also been described in [2].

The results from [14] and [2] allow us to extend our results to tree embeddings with additive distortion guarantees. The results described above directly lead to k-points condition for additive distortion.

THEOREM 4.2. *Every k-point submetric of a metric space (X, d) embeds into a tree with additive distortion $\mathrm{O}(\delta)$ if and only if for every $r \in X$ and for every sequence $x = x_1, \ldots, x_t = y$ of length $t < k$, $(x|y)_r \geq \min_{1 \leq i < t}(x_i|x_{i+1})_r - \mathrm{O}(\delta)$.*

For $k = 4$, the k-points condition of Theorem 4.2 is equivalent to the δ-hyperbolicity condition (Definition 1.3). In [14] it is shown that for δ-hyperbolic metrics (X, d), $(x|y)_r \geq \min_{1 \leq i < t}(x_i|x_{i+1})_r - \delta \cdot \lceil \log |X| \rceil$ for every $r \in X$ and every sequence of points $x_i \in X \setminus \{r\}$. The argument used in [14] can be extended to the k-points condition given by Theorem 4.2 and we obtain the following theorem.

THEOREM 4.3. *If every k-point submetric of a metric space (X, d) embeds into a tree metric with additive distortion δ, then the metric (X, d) embeds into a tree metric with additive distortion $\mathrm{O}(\delta \cdot \log_k |X|)$.*

By using a slight variation of the pyramid construction of Definition 3.1, we can also extend the lower bound of Section 3 to a similar statement about additive distortion.

THEOREM 4.4. *There is a metric space (X, d) such that every submetric induced by k point of (X, d) embeds into a tree metric with additive distortion δ but where every tree embedding of (X, d) requires additive distortion $\Omega(\delta \cdot \log_k |X|)$.*

5. REFERENCES

[1] I. Abraham, Y. Bartal, and O. Neiman. Embedding metrics into ultrametrics and graphs into spanning trees with constant average distortion, 2006.

[2] R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM Journal on Computing*, 28(3):1073–1085, 1999.

[3] S. Arora, L. Lovász, I. Newman, Y. Rabani, Y. Rabinovich, and S. Vempala. Local versus global properties of metric spaces. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 41–50, New York, NY, USA, 2006. ACM Press.

[4] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science (Burlington, VT, 1996)*, pages 184–193. IEEE Comput. Soc. Press, Los Alamitos, CA, 1996.

[5] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak. Operating System Support for Planetary-Scale Network Services. In *Proc. of the Symposium on Networked Systems Design and Implementation*, pages 253–266, San Francisco, CA, Mar. 2004.

[6] E. Begelfor and M. Werman. The world is not always flot or learning curved manifolds. Technical Report 65, School of Engineering and Computer Science, Hebrew University of Jerusalem, 2005.

[7] P. Buneman. A note on metric properties of trees. *Journal of Combinatorial Theory, Ser. B*, 17:48–50, 1974.

[8] M. Costa, M. Castro, A. Rowstron, and P. Key. Pic: Practical internet coordinates for distance estimation. In *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pages 178–187, Washington, DC, USA, 2004. IEEE Computer Society.

[9] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 15–26, New York, NY, USA, 2004. ACM Press.

[10] M. Elkin, Y. Emek, D. A. Spielman, and S.-H. Teng. Lower-stretch spanning trees. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 494–503, New York, NY, USA, 2005. ACM Press.

[11] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 448–455. ACM Press, 2003.

[12] C. Gavoille and O. Ly. Distance labeling in hyperbolic graphs. In 16^{th} *Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 3827 of Lecture Notes in Computer Science, pages 1071–1079. Springer, Dec. 2005.

[13] C. Gavoille, D. Peleg, S. Pérenneès, and R. Raz. Distance labeling in graphs. *Journal of Algorithms*, 53(1):85–112, 2004.

[14] M. Gromov. Hyperbolic groups. *Essays in Group Theory*, pages 75–264, 1987.

[15] A. Gupta. Steiner points in tree metrics don't (really) help. In *SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 220–227, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[16] P. Indyk. Algorithmic applications of low-distortion geometric embeddings. In *FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 10, Washington, DC, USA, 2001. IEEE Computer Society.

[17] J. M. Kleinberg, A. Slivkins, and T. Wexler. Triangulation and embedding using small sets of beacons. In *FOCS*, pages 444–453, 2004.

[18] R. Krauthgamer and J. R. Lee. Algorithms on negatively curved spaces. In *Foundations of Computer Science, 2006. FOCS '05. 47th Annual IEEE Symposium on*, pages 119–132, Berkeley, CA, USA,, Oct. 2006.

[19] H. Lim, J. Hou, and C.-H. Choi. Constructing internet coordinate system based on delay measurement. In *Proc. of ACM/SIGCOMM Internet Measurement Conference (IMC)*, Oct. 2003.

[20] N. Linial. Finite metric spaces–combinatorics, geometry and algorithms. Proceedings of the ICM, Beijing 2002, vol. 3, 573–586.

[21] J. Matoušek. Open problems on low-distortion embeddings of finite metric spaces, 2005. Avaliable in http://kam.mff.cuni.cz/ matousek/metrop.ps.gz.

[22] R. Mihaescu, D. Levy, and L. Pachter. Why neighbor-joining works, 2006.

[23] T. S. E. Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *INFOCOM*, 2002.

[24] D. Peleg. Proximity-preserving labeling schemes. *Journal of Graph Theory*, 33:167–176, 2000.

[25] M. Pias, J. Crowcroft, S. Wilbur, S. Bhatti, and T. Harris. Lighthouses for scalable distributed location. In *Proc. of International Workshop on Peer-to-Peer Systems (IPTPS)*, Feb. 2003.

[26] Y. Rabinovich and R. Raz. Lower bounds on the distortion of embedding finite metric spaces in graphs. *Discrete & Computational Geometry*, (19), 1998.

[27] N. Saitou, M. Nei, et al. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol*, 4(4):406–425, 1987.

[28] Y. Shavitt and T. Tankel. On the curvature of the internet and its usage for overlay construction and distance estimation. In *Proc. of the 23th Ann. INFOCOM*, 2004.

[29] L. Tang and M. Crovella. Virtual landmarks for the internet. In *Proc. of ACM/SIGCOMM Internet Measurement Conference (IMC)*, Oct. 2003.

[30] L. wei Lehman and S. Lerman. Pcoord: Network position estimation using peer-to-peer measurements. In *Proc. of IEEE International Symposium on Network Computing and Applications (NCA)*, Aug. 2004.

[31] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: a lightweight network location service without virtual coordinates. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 85–96, New York, NY, USA, 2005. ACM Press.

[32] C. Yoshikawa. All-sites-pings for planetlab. `http://ping.ececs.uc.edu/ping/`, 2006.